

AD-A135 232

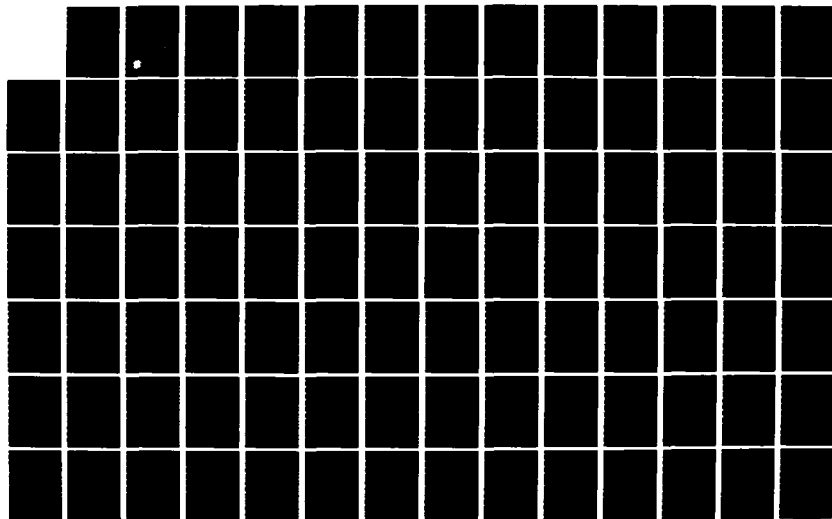
REAL-TIME IMPLEMENTATION OF A SPEECH DIGITIZATION  
ALGORITHM COMBINING TIM. (U) NOTRE DAME UNIV IN DEPT OF  
ELECTRICAL ENGINEERING J L MELSA ET AL. JUN 83  
DCA100-82-C-0026

1/2

UNCLASSIFIED

F/G 9/4

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

13

FINAL REPORT - VOLUME 1  
REAL-TIME IMPLEMENTATION  
OF A  
SPEECH DIGITIZATION ALGORITHM  
COMBINING  
TIME-DOMAIN HARMONIC SCALING  
AND  
ADAPTIVE RESIDUAL CODING

DEC 2 1983  
A

Department of

ELECTRICAL ENGINEERING

UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA



DTIC FILE COPY

AD-A135232

This document has been approved  
for public release and sale; its  
distribution is unlimited.

83 12 02 028

**FINAL REPORT - VOLUME 1**  
**REAL-TIME IMPLEMENTATION**  
**OF A**  
**SPEECH DIGITIZATION ALGORITHM**  
**COMBINING**  
**TIME-DOMAIN HARMONIC SCALING**  
**AND**  
**ADAPTIVE RESIDUAL CODING**

This document has been approved  
for public release and sale; its  
distribution is unlimited.

Prepared for

Defense Communications Agency  
Defense Communications Engineering Center  
1860 Wichle Avenue  
Reston, Virginia 22090

Contract No. DCA-100-82-C-0026

1 June 1983

DTIC  
ELECTE  
DEC 2 1983  
A

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  REAL-TIME IMPLEMENTATION OF A SPEECH DIGITIZATION ALGORITHM COMBINING TIME-DOMAIN HARMONIC SCALING AND ADAPTIVE RESIDUAL CODING		5. TYPE OF REPORT & PERIOD COVERED  Final Feb. 1982 to May 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)  James L. Melsa James D. Mills Arvind A. Arora		8. CONTRACT OR GRANT NUMBER(s)  DCA-100-82-C-0026
9. PERFORMING ORGANIZATION NAME AND ADDRESS  Department of Electrical Engineering University of Notre Dame Notre Dame, IN 46556		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS  Defense Communications Agency Contract Management Division, Code 260 Washington, DC 20305		12. REPORT DATE  June 1983
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)  Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Distribution of this document is unlimited. If may be released to the Clearinghouse, Department of Commerce, for sale to the public.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  Speech coding, 16000 bps speech transmission, time-domain harmonic scaling, adaptive residual coding, hardware realization.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  This report describes the results of a fifteen-month study, supported by DCA Contract 100-82-C-0026, of the real-time implementation of algorithm com- bining time-domain harmonic scaling (TDHS) and Adaptive Residual Coding (ARC) at a transmission bit rate of 16 kb/s. The modifications of this encoding algorithm as originally presented by Melsa and Pande (1981) to allow real-time implementation are described in detail. A non real-time FORTRAN simulation		

(continued)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

using a sixteen-bit word length was developed and tested to establish feasibility. The hardware implementation of a full-duplex, real-time system has demonstrated that this algorithm is capable of producing toll quality speech digitization.

This report has been divided into two volumes. The first volume discusses the algorithm modifications and FORTRAN simulation. The details of the hardware implementation, schematics for the system and operating instructions are included in Volume 2 of this final report.

Unclassified

## ABSTRACT

This report describes the results of a fifteen-month study, supported by DCA Contract 100-82-C-0026, of the real-time implementation of algorithm combining time-domain harmonic scaling (TDHS) and Adaptive Residual Coding (ARC) at a transmission bit rate of 16 kb/s. The modifications of this encoding algorithm as originally presented by Melsa and Pande (1981) to allow real-time implementation are described in detail. A non real-time FORTRAN simulation using a sixteen-bit word length was developed and tested to establish feasibility. The hardware implementation of a full-duplex, real-time system has demonstrated that this algorithm is capable of producing toll quality speech digitization.

This report has been divided into two volumes. The first volume discusses the algorithm modifications and FORTRAN simulation. The details of the hardware implementation, schematics for the system and operating instructions are included in Volume 2 of this final report.



Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

PROJECT PERSONNEL

James L. Melsa, principal investigator

James D. Mills, research assistant

# TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT .....	11
PROJECT PERSONNEL .....	111
LIST OF FIGURES .....	vi
LIST OF ABBREVIATIONS .....	vii
CHAPTER I: INTRODUCTION .....	1
CHAPTER II: ORIGINAL SYSTEM .....	3
2.1 Introduction .....	3
2.2 Transmitter .....	6
2.3 Channel .....	7
2.4 Receiver .....	8
CHAPTER III: FINAL SYSTEM .....	10
3.1 Introduction .....	10
3.2 Transmitter .....	10
3.3 Channel .....	13
3.4 Receiver .....	13
CHAPTER IV: INITIAL ALGORITHM REFINEMENTS .....	15
4.1 Introduction .....	15
4.2 16 kb/s .....	15
4.3 Bit error rate .....	24
CHAPTER V: HARDWARE INDUCED MODIFICATIONS .....	28
5.1 Introduction .....	28
5.2 Serial Pitch Extraction .....	28
5.3 8 kHz Sampling Rate .....	36
5.4 Hardware Compatible ARC .....	37
5.5 Finite Word Length .....	39
CHAPTER VI: METHODOLOGY .....	41
6.1 Introduction .....	41
6.2 Files and Options .....	41
CHAPTER VII: CONCLUSIONS .....	49
APPENDIX A: FINITE WORD LENGTH SIMULATION SOURCE FILES .....	51

	<u>Page</u>
APPENDIX B: UNRESTRICTED WORD LENGTH SOURCE FILES .....	111
APPENDIX C: SUPPORTING PROGRAMS .....	160
APPENDIX D: FILTER DESIGN .....	171
APPENDIX E: FINAL OUTPUT EXAMPLES .....	172
REFERENCES: .....	180

## LIST OF FIGURES

		<u>Page</u>
Figure 2.1.1	Original TDHS-ARC System .....	4
Figure 2.1.2	Original System Basic Structure .....	5
Figure 3.1.1	Final TDHS-ARC System .....	11
Figure 4.1.1	First Stage of Algorithm Refinements .....	16
Figure 4.2.1	Determination of Scaling Factors and Thresholds .....	18
Figure 4.2.2	Hand Picked Scaling Factors and Thresholds .....	22
Figure 4.2.3	Logarithmically Distributed Scaling Factors and Thresholds .....	23
Figure 4.2.4	Quantizer Thresholds Scaling Factors, and Expansion Factors for Speech at 6400 Hz and 8000 Hz Sampling .....	25
Figure 4.3.1	Channel with Temporary BER Assumption .....	27
Figure 5.2.1	TDHS Algorithms .....	31
Figure 5.2.2	Center Clipping Algorithms .....	33
Figure 6.1.1	Final System Algorithm and Output Files ...	42
Figure 6.2.1	Graphical Pitch Extraction .....	44
Figure 6.2.2	Algorithm Capabilities .....	47

## LIST OF ABBREVIATIONS

DCA	Defense Communications Agency
TDHS	time domain harmonic scaling
ARC	adaptive residual coder
FWL	finite word length
TDHC	time domain harmonic compression
TDHE	time domain harmonic expansion
CVSD	continuously variable slope delta modulator
$s(k)$	input sampled speech signal
$y(k)$	compressed speech signal
$q(k)$	quantizer level sequence
$\hat{q}(k)$	system estimate of the original $q(k)$
$\hat{y}(k)$	system estimate of the original $y(k)$
$\hat{s}(k)$	system estimate of the original $s(k)$
LPF	low pass filter
FIR	finite impulse response
Hz	Hertz
kb/s	kilobits per second
b/s	bits per second
BER	bit error rate
$\Delta$	triangular or change in
NQ	number of quantization level
MSF	maximum scaling factor
$sf_q$	$q^{\text{th}}$ scaling factor
$TH_q$	$q^{\text{th}}$ threshold

$e(k)$	error sequence
$\hat{e}(k)$	system estimate of original $e(k)$
$\sigma(k)$	approximation to the standard deviation of input sequence
dB	decibels
$C_k$	clipping level
$E_k$	approximation of the energy

## CHAPTER I

### INTRODUCTION

This report describes the results of a fifteen-month study, supported by DCA Contract 100-82-C-0026, of the real-time implementation of an algorithm combining time-domain harmonic scaling (TDHS) and Adaptive Residual Coding (ARC) at a transmission bit rate of 16 kb/s. The modifications of this encoding algorithm as originally presented by Melsa and Pande (1981) to allow real-time implementation are described in detail. A non real-time FORTRAN simulation using a sixteen-bit word length was developed and tested to establish feasibility. The hardware implementation of a full-duplex, real-time system has demonstrated that this algorithm is capable of producing toll quality speech digitization.

This report has been divided into two volumes. The first volume discusses the algorithm modifications and FORTRAN simulation. The details of the hardware implementation, schematics for the system and operating instructions are included in Volume 2 of this final report.

The minimum requirements for the TDHS-ARC speech digitization technique are that it should provide toll quality speech under optimum conditions and high quality robust speech in the presence of 60 dB acoustic background noise and in the presence of random channel error rates up to 0.001. The algorithm should be designed for a data rate of 16 kb/s.

[Statement-of-work]

The trade-off between transmission quality and hardware complexity was studied with the overall objective of finding algorithm refinements that simplify implementation without degrading the previous performance.

It is important to note that minor sacrifices in system performance may be made to achieve significant hardware simplification. This volume presents the software simulation for a time domain harmonic scaling - adaptive residual coder system. It includes the finite word length simulations as the final form.

A necessary condition for understanding the direction and significance of this work is a thorough familiarity with the original form of the TDHS-ARC 9.6 kb/s system. Therefore, a brief survey of this system comprises the second chapter. Under the heading of "Final System", the third chapter outlines the final algorithm. This chapter will highlight the major modifications without providing the background associated with each variation. The fourth chapter presents the algorithm changes which were primarily due to the system options called for in the statement of work. The changes dictated by hardware considerations are discussed in the fifth chapter. The sixth chapter reports the effects of replacing the error-free channel approximation with a noisy channel. The seventh chapter is concerned with an explanation of the program options and output files, as well as their use. The final chapter forms a summary of the work and the final system.

## CHAPTER II

### ORIGINAL SYSTEM

#### 2.1 Introduction

Throughout this report references will be made to the "original system". This is a reference to the 9.6 kb/s TDHS-ARC system presented by Pande [Melsa and Pande, 1981]. A block diagram which reflects the major points of interest of the system on this date is illustrated in Fig. 2.1.1, entitled "Original TDHS-ARC System". This is essentially the beginning point of the research herein reported. In order to evaluate the significance of this work, it is necessary to understand this original system. This chapter also highlights the characteristics of the original system which were modified to obtain the final system; these modifications are the subjects of subsequent chapters.

The basic structure of the original system is illustrated in Fig. 2.1.2. This structure was first proposed by Malah in 1980 when he presented the results of a TDHS-CVSD system and suggested the usefulness of combining TDHS with other waveform coders [Malah, 1980]. The structure is a time domain harmonic compression unit concatenated with an adaptive residual encoder in the transmitter, and an adaptive residual decoder concatenated with a time domain harmonic expansion unit in the receiver. The receiver's components perform the inverse function of the transmitter's sections (logically resulting in receiver expressions such as inverse ARC). The quality is limited by noise sources, such as, quantization in the ARC and compression versus expansion differences in TDHS. Again, it is worthwhile to note that this structure is the heart of the system and was maintained throughout the investigations.

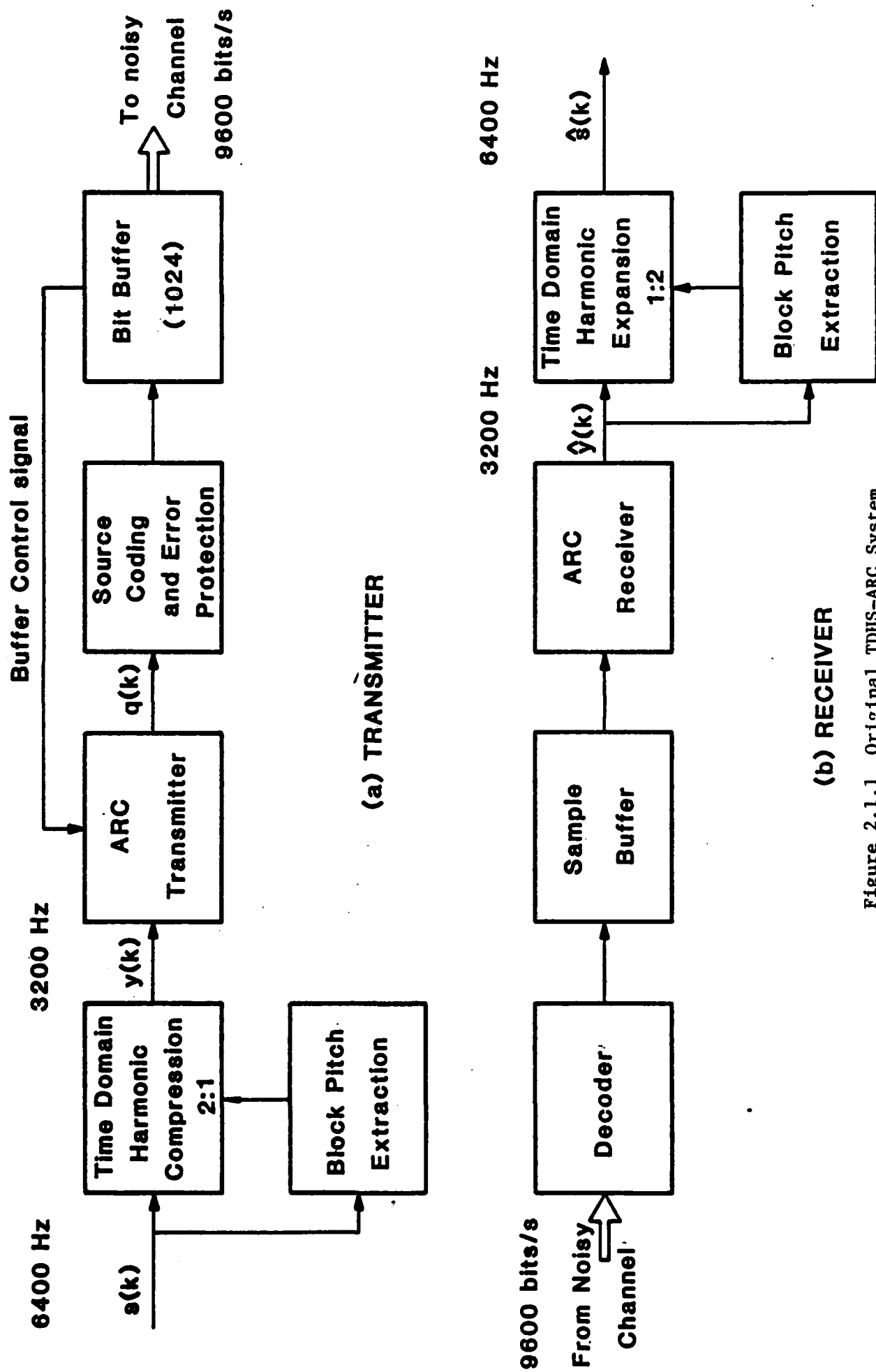


Figure 2.1.1.1 Original TDHS-ARC System



Figure 2.1.1.2 Original System Basic Structure

The many notable characteristics of the original system are identifiable as pertaining to the transmitter, channel, or receiver. They have been individually classified as such and are presented in the following sections. Many of the points of interest of the receiver are similar to those of the transmitter. They are, therefore, noted as being congruent and only briefly mentioned.

## 2.2 Transmitter

The input to the transmitter, in Fig. 2.1.1 is denoted  $s(k)$ , an analog speech signal sampled at 6400 Hz. This sequence forms the input to both the compression block and the block pitch extraction unit which jointly form the TDHC algorithm in the transmitter. A compression ratio of 2:1 is used and although many acceptable\* windows may be used, a triangular window was suggested due to computational simplicity [Melsa and Pande, 1981]. A higher compression ratio can be effective. However, the choice of a higher scaling factor entails the need for a more complex window function for equivalent performance [Malah, 1981]. The new characterization of block pitch extraction was precipitated by a major topic of investigation which is described in Chapter V. A pitch period is determined for a block of speech by locating the maximum of a correlation computation. The block length and range of the computation are 100 and 20 to 100 respectively. The output of the TDHE unit is a compressed 3200 samples/second speech signal.

---

\* An acceptable window is defined as any window  $h(1:N_p)$  such that

$$\begin{aligned} h(1:N_p) &= 1 \\ h(N_p:N_p) &= 0 \end{aligned}$$

This is required by continuity considerations.

The compressed signal constitutes the input to the transmitter ARC. The encoder embodies a fourth order adaptive predictor and a 21 level buffer controlled quantizer. The adaptation of the sequentially adaptive linear predictor coefficients is based upon the steepest descent of the energy in the error between the predicted and actual values [Melsa and Pande, 1981]. The buffer control signal shifts the quantizer thresholds to reduce the possibility of buffer underflow or overflow. This is achieved by forcing or restricting the occurrence of a run length codeword. The sequence of quantization levels is the only information passed on to the remainder of the transmitter, the channel, and eventually, the receiver. The receiver must, therefore, require no more than this information.

The coding block accepts the quantization levels and maps them to a variable length codeword sequence that adds the error protection. The variable length code precipitates the need for a buffered output to the channel. The buffer serves a dual role. It allows the transmitter to emit bits at a constant rate and, as illustrated by the buffer control signal, it presents the adaptation information to the variable quantizer. The buffer outputs bits to the channel at a transmission rate of 9600 bits per second.

### 2.3 Channel

The original system incorporated a noisy channel simulation with an interactive bit error rate. The channel model uses a pseudo-random number generator to determine the location of a bit error in the channel sequence. Given the decision of the pseudo-random number generator, the algorithm changes the bit stream accordingly.

The channel is constrained to a 9600 bits/second rate. Although this is a channel constraint, its influence is accommodated in the receiver and transmitter algorithms by restricting the average bits allowed per quantization level. In this way, it affects the coding and decoding blocks and the fineness of quantization in both ARC algorithms.

Therefore, the importance of the simulated channel lies in its ability to induce errors in the bit stream between the transmitter and receiver. The transmission rate of 9600 bits/second plays a role in the transmitter and receiver algorithms and not in the channel algorithm.

#### 2.4 Receiver

A bit stream at the transmission rate of 9600 bit/second is received from the channel by the decoder. The decoding block maps the bit stream back into a sequence of quantization levels which may or may not deviate from the original quantizer level sequence. The efficiency of mapping  $q(k)$  to  $\hat{q}(k)$  is dependent only upon the specified error correcting capability, bit error rate, and code. Again, the requirement of a sample buffer is generated by the utilization of a variable length code. The sample buffer contains the quantization levels contaminated by noise induced in the channel.

These levels are accessed by the inverse ARC block. This ARC is also a fourth order adaptive linear predictor combined with a 21 level quantizer. Note that there is no need for a buffer control signal in the receiver and that the receiver need not know the threshold values used by the transmitter. Other than these major disparities, the inverse ARC operates as the exact inverse to the encoder in the transmitter.

The time domain harmonic expansion element in the receiver acts, as stated earlier, much the same as the compression blocks. A pitch extraction routine identical to that in the transmitter, is exploited in the receiver. The major relevant differences are the expansion ratio, which is, of course, 1:2, the exact inverse of the compression ratio, and the window. Here, the choice of windows is forced due to the choice of a triangular window in the transmitter. Once again employed is the notion of an acceptable window. That is, one that maintains the continuity at its fringes, as well as those of the succeeding window. The impelling concept of this algorithm, as a whole, is the production of a high quality sequence of estimates,  $\hat{s}(k)$ , to the original speech sequence,  $s(k)$ . This estimate forms the output of the receiver and hence the algorithm. It is considered a 6400 Hz sampled speech signal estimate.

## CHAPTER III

### FINAL SYSTEM

#### 3.1 Introduction

A block diagram of the final system is illustrated in Fig. 3.1.1. This simulation incorporates the specifications cited in Chapter I. It is immediately apparent that the final algorithm maintained the basic structure discussed in the opening remarks of Chapter II and illustrated in Fig. 2.1.2. The transmitter preserves the TDHC-ARC format and the receiver upholds the inverse ARC-TDHE structure. Given that the major concept of a TDHE ARC system is maintained, the points of interest of the final algorithm are the modification of the individual components. Although detailed discussions of these changes are deferred for future chapters, they are highlighted in the remainder of this chapter.

#### 3.2 Transmitter

The input to the transmitter is composed of an analog speech signal sampled at 8000 Hz. This signal forms the input to both the compression block and the pitch extraction block. The compression ratio of 2:1, is preserved in this final algorithm. Also preserved is the utilization of a triangular window, the length of which is twice the detected pitch period length.

The pitch extraction block underwent crucial transformations. The input sampled speech is first low-pass filtered by a 9th order linear phase FIR digital filter before pitch detection is performed. Following this operation, a serial pitch extraction routine is executed. The investigation of this routine constituted the majority of the effort

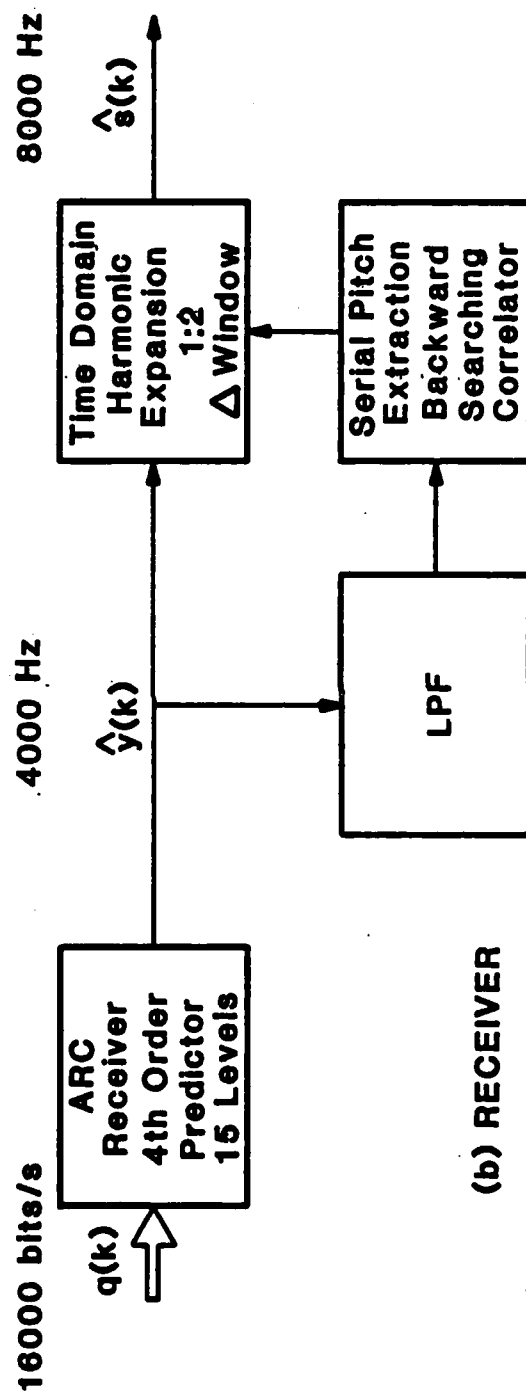
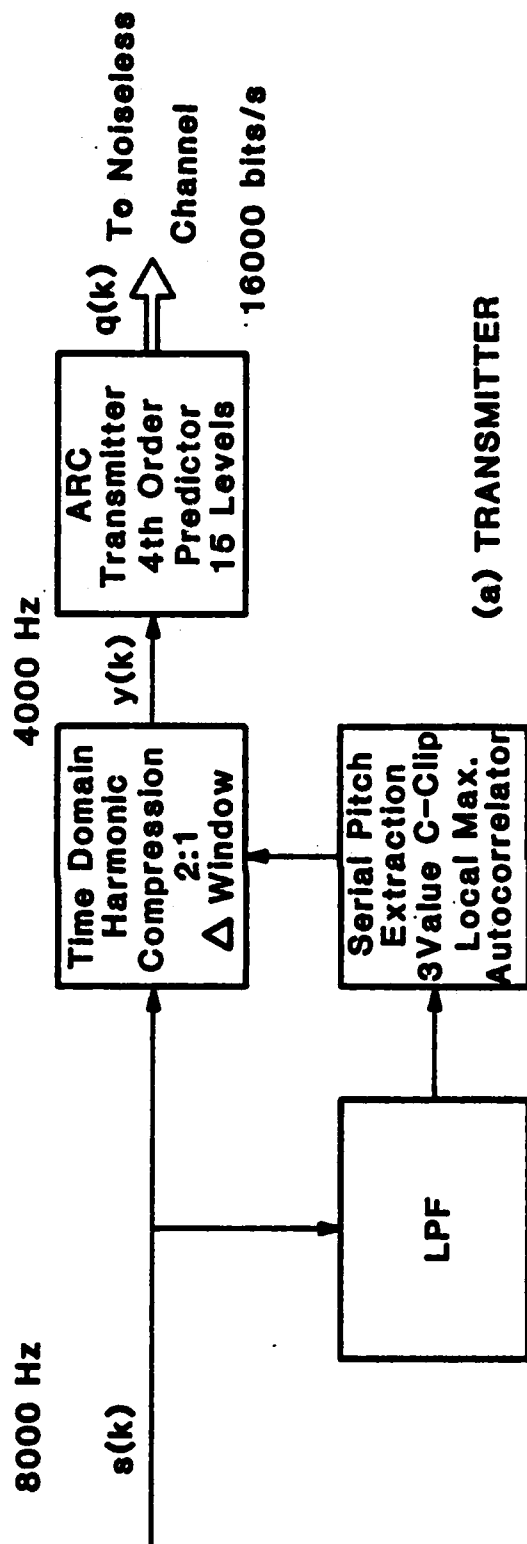


Figure 3.1.1 Final TDHS-ARC System

allotted to the modification of the TDHS algorithm. It is based on hardware considerations presented in Chapter V. Pseudo-random pitch picking and a new processing of the correlation function's data are exercised by the changes. The hardware simplicity associated with the three value center clipped autocorrelation computation was retained.

The adaptive residual encoder found in the transmitter incorporated a fifteen level fixed quantizer and a fourth order linear predictor. The adaptation procedure is modified to eliminate the division operation called for in the original system. This is accomplished by adopting a fixed gain adaptation procedure. Another division operation is eliminated by using the quantizer normalization factor as a multiplicative adaptation to the thresholds and scaling factors as opposed to a dividing adaptation to the error signal. This results in only a mild truncation of the original system's performance. The division operations are eliminated to make the algorithm hardware compatible with the NEC7720 signal processing chip which does not support a division operation. The use of a fixed length code, a sampling rate of 8000 Hz and the restriction to a 16 kilobit/second transmission rate dictated that the maximum number of quantization levels be 15.

Due to algorithm changes presented in Chapter IV, the quantization levels became the information passed from the transmitter to the receiver. The 15 quantization levels require 4 transmission bits/sample resulting in an effective transmission rate of 4000 quantization level samples/second. The quantization levels, therefore formed the output of the transmitter.

### 3.3 Channel

A temporary approximation to a noiseless channel is instrumental in the reduction of the transmitter and receiver algorithms to their form in Fig. 3.1.1. The performance of the system in a noisy channel is reported in Chapter VI. A crucial consequence of the noiseless channel approximation is that the output of the transmitter is exactly the input to the receiver. In essence, the channel is not present, save the transmission rate characteristic.

The channel transmission rate of 16 kilobits/second is another of the channel's features that dictated significant alterations to the original system. Although these variations did not predominate in the channel model itself, they are critical to the final algorithm. The transmission rate restricted the quantization of the ARC and thus, upper-bounded the signal-to-noise ratio of the ARC system. The effects of transmission rate, then, are sustained in the transmitter and receiver and not in the channel.

### 3.4 Receiver

The channel output, which exactly coincides with its input and the transmitter output, becomes the input to the receiver. It is considered a 16 kilobit/second signal representing 4 bit codewords of the quantization levels. Since the quantizer levels form the transferred information, no decoding is necessary and inverse ARC is applied directly to the input. Inverse ARC in this system is equivalent to the feedback loop of the ARC in the transmitter. It contains a fourth order linear predictor as in the original system. The quantizer has 15 levels and is

fixed in the final algorithm as opposed to the 21 adaptive levels used in the original. The attractive characteristic of the inverse ARC (not requiring the threshold values) is maintained in the final algorithm.

The compressed signal  $\hat{y}(k)$  is a 4000 Hz sequence. It is the estimate of the compressed speech signal in the transmitter,  $y(k)$ , and is the driving sequence of the expansion and pitch extraction units of the receiver. The expansion block of the receiver behaves exactly as that of the original system. This block perpetuates the 1:2 expansion ratio and the triangular window suggested by the original. Again, research activities concentrated on the pitch extraction algorithm. The advent of serial pitch extraction and its inclusion and success in the transmitter ordained its use in the receiver.

In previously investigated algorithms, the pitch extraction algorithm of the receiver was identical to that of the transmitter. An important innovation of this final algorithm is that there exists a major difference in the receiver's algorithm as compared to the transmitter's. Noting that the receiver has a reduced amount of pitch information available to it, the receiver algorithm is modified to more judiciously use this information. The receiver correlates backward as well as forward one searching blocklength for pitch information. The transmitter algorithm only looks forward.

The product of the receiver is a sequence,  $\hat{s}(k)$ , which is the approximation of the input speech sequence,  $s(k)$ . The rate is equal to the input rate of 8000 Hz. The similarity  $\hat{s}(k)$  and  $s(k)$  is used as an indication of the performance of the system. Therefore, side-by-side taped comparisons are used as a major indication of performance.

## CHAPTER IV

### INITIAL ALGORITHM REFINEMENTS

#### 4.1 Introduction

The initial algorithm refinements are described in this chapter. These refinements are those that are principally required by the mandates of Chapter I. The hardware-induced modifications are deferred to Chapter V. Chapter I imposes two considerable requirements. It requests a transmission rate of 16000 bits/second and a bit error rate of 0.1%. These conditions immediately translate to significant algorithm adjustments. Although both stipulations are channel characteristics, it is necessary to modify the transmitter and receiver algorithms since the channel is considered unalterable.

The system, as it appeared following these initial changes, is illustrated in Fig. 4.1.1. Hardware induced refinements that arose after these initial changes, such as the switch to 8 kHz sampled input speech, are considered in Chapter V. The discussion of their effects on the refinements presented in this chapter are also deferred to Chapter V. The following discussion is subdivided into two sections. The first describes the refinements due to the new transmission rate. The second relates the added alterations caused by the BER condition.

#### 4.2 16 kb/s

The 16 kb/s transmission rate and the use of a 6400 Hz sampled speech signal affected the ARC and inverse ARC considerably. The high transmission rate and the low sampling rate allowed the use of a fixed length (non-run length) code. A bit rate of 16 kb/s and compressed

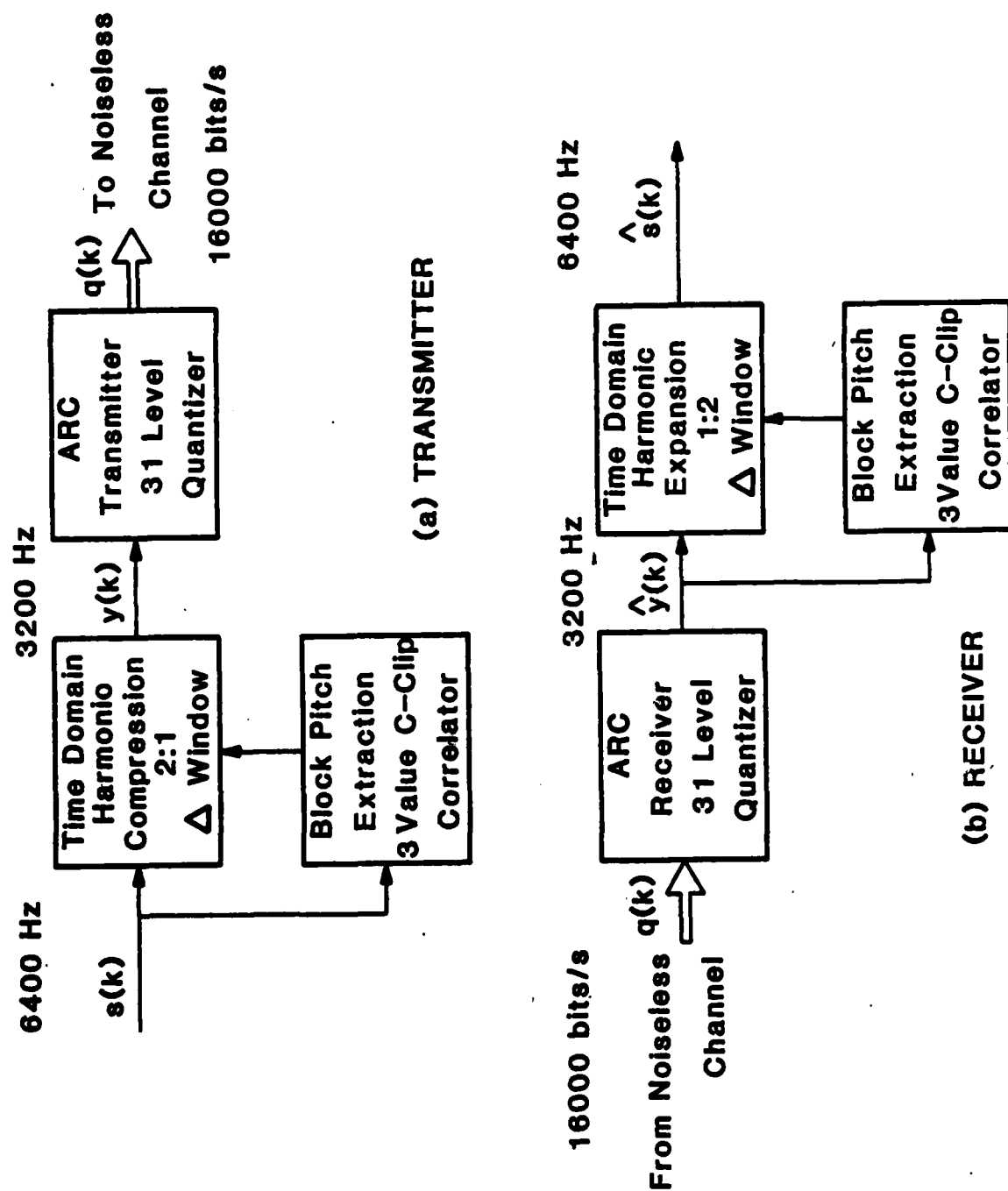


Figure 4.1.1.1 First Stage of Algorithm Refinements

speech rate of 3200 Hz entitles the algorithm to an average of 5 bits per sample. A five bit fixed length code allows for a 32 level quantizer. Since the quantizer is symmetric about its center, an odd number of quantizer levels are used. Therefore, the sampling frequency and transmission rate dictate a 31 level quantizer if a fixed code is used. Since an eleven level quantizer produced good results [Pande, 1981] in the original algorithm a 31 level quantizer is initially accepted (and experimentally supported in subsequent tests).

Earlier investigations involved fewer levels (11, for instance). In these investigations, the thresholds, scaling factors, and expansion factors are judiciously hand-picked. (The thresholds and scaling factors are the input and output values of the quantizer and the expansion factors are the scalars used to update the standard deviation computation.) The advent of 31 levels brings with it the need for a more systematic method of choosing these values. Because the magnitudes of speech samples are inherently exponentially distributed, a method which incorporated a logarithmic scaling of the quantizer levels was pursued. The final system allows for the choice of a base and a maximum scaling factor. The method is illustrated in Fig. 4.2.1. In this figure, the following abbreviations and definitions are employed

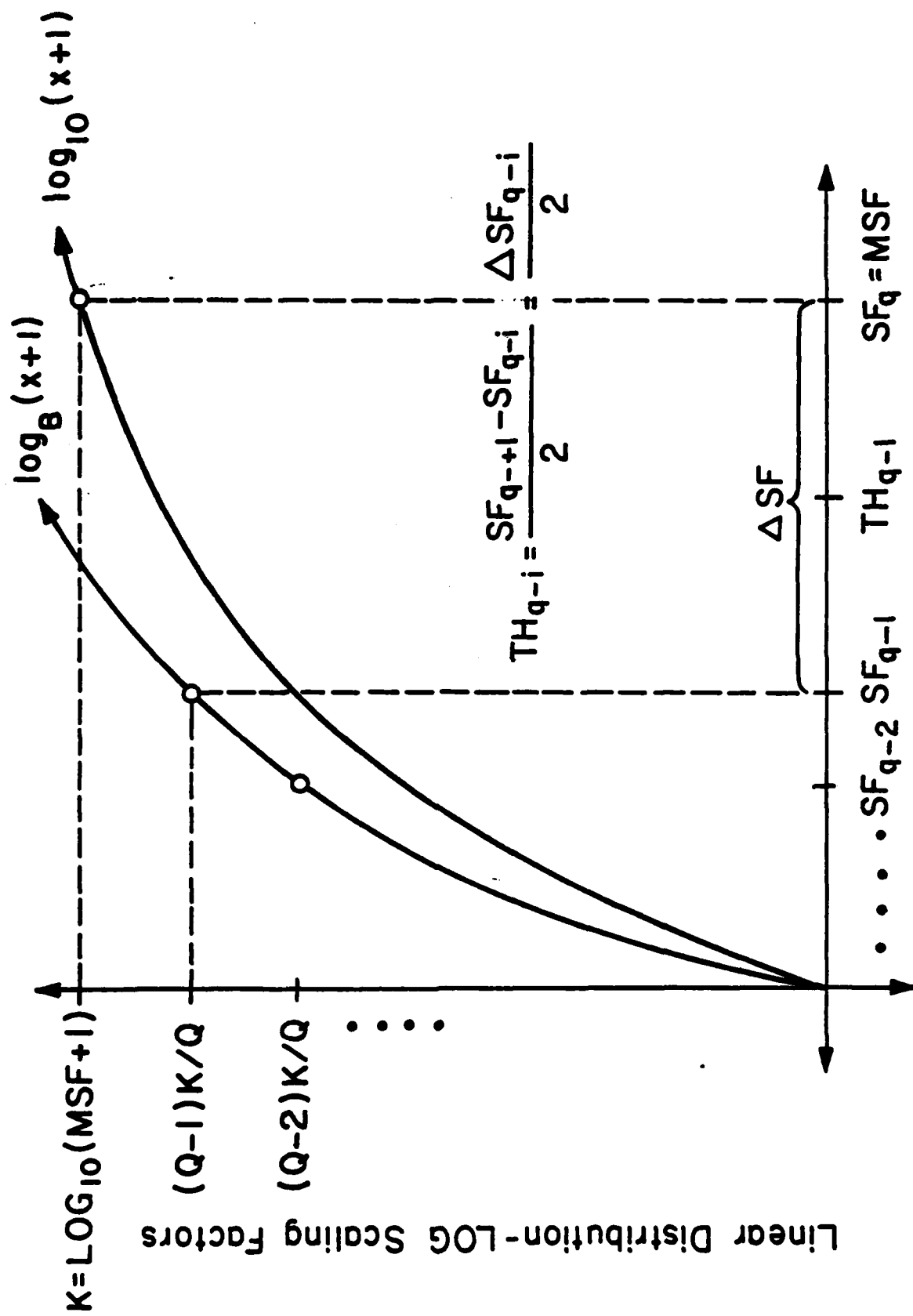
NQ = number of quantization levels,

MSF = maximum scaling factor,

SF<sub>q</sub> = q<sup>th</sup> scaling factor, and

TH<sub>q</sub> = q<sup>th</sup> threshold.

The quantizer is symmetric about zero because speech is nearly as



Generated Scaling Factors and Thresholds

# DETERMINATION OF SCALING FACTORS AND THRESHOLDS

negative-going as it is positive-going. Therefore, the zero-level is pre-chosen and there are only Q independent levels, where

$$Q = \frac{NQ-1}{2} .$$

The maximum scaling factor is mapped to the value K on the ordinate by the shifted logarithmic function

$$Y = \log_{10}(X+1) .$$

(X being the independent variable.) The ordinate is then equally divided into Q sections from 0 to K where

$$K = \log_{10}(MSF+1) .$$

The ordinate values are, therefore,

$$\begin{array}{c} 0 \\ (1/Q)K \\ (2/Q)K \\ \vdots \\ ((Q-2)/Q)K \\ ((Q-1)/Q)K \\ K \end{array}$$

It is desired to map these ordinate values back to the abscissa to obtain the logarithmically scaled values. To add flexibility to this technique, the base of the logarithm used to map the values from the ordinate to the abscissa, termed the distributing base, was made

variable. The change in base from 10 to the chosen B allows the lowest Q quantizer levels to be expanded away from or contracted toward the origin. It was noted that if the distributing base was chosen to be too large, the second largest ordinate value would be mapped to a value that exceeds the requested maximum scaling factor.

An upper bound on the distributing base is determined. The bound is resolved using the constraint that the second largest ordinate value cannot be mapped back to a value that exceeds the requested maximum scaling factor. The bound is

$$\text{Base} \leq 10 \exp \left\{ \frac{NQ}{NQ-1} [\log(\text{MSF}+1)/\log(\text{MSF})] \right\}$$

A distributing base, B, is then chosen and the function

$$Y = \log_B (X+1)$$

is used to map the Q-1 unspecified scaling factors back to the abscissa. Here, Y would be the independent variable and in the usual presentation, the function is written

$$X = (B^Y - 1)$$

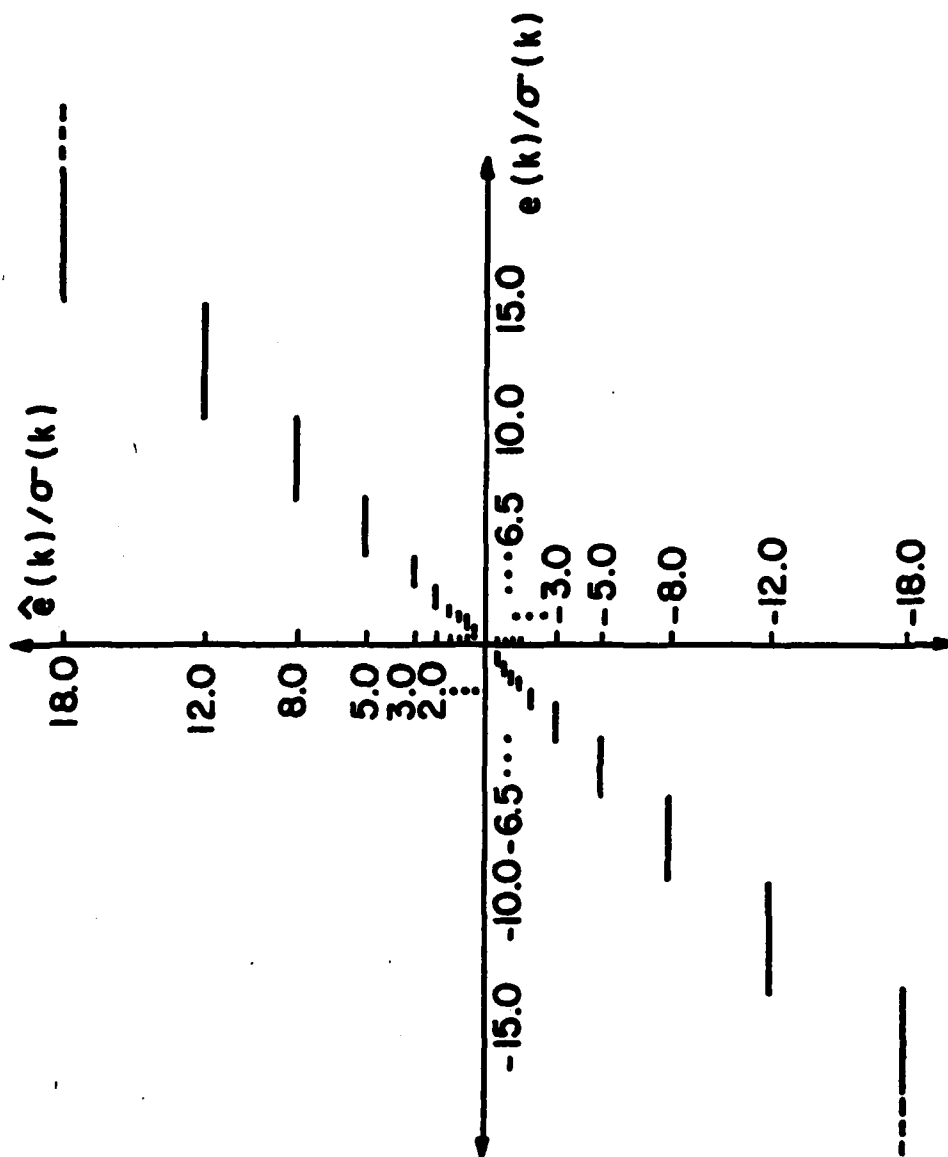
A shifted logarithmic scale is exploited to force the second function to map the scaling factors logarithmically from the zero level to the second largest scaling factor. The largest ordinate value, K, is mapped back to the abscissa using the original base (=10) so that the specified maximum scaling factor is maintained.

The effectiveness of this technique is justified by a comparison between the methodically chosen factors and the hand-picked factors. A minimum mean square error is used to compare the two sets of data. Other comparisons, such as percent error were considered and discarded. Their undesirable weighting of small errors in the lower scaling factors led to their disqualification. Using the mean square error criterion, a local minimum mean square error was found to be 0.4 and occurred at  $B = 9.0$ .

As in preceeding work [Melsa and Pande, 1981], the thresholds were chosen to be equidistant from successive scaling factors.

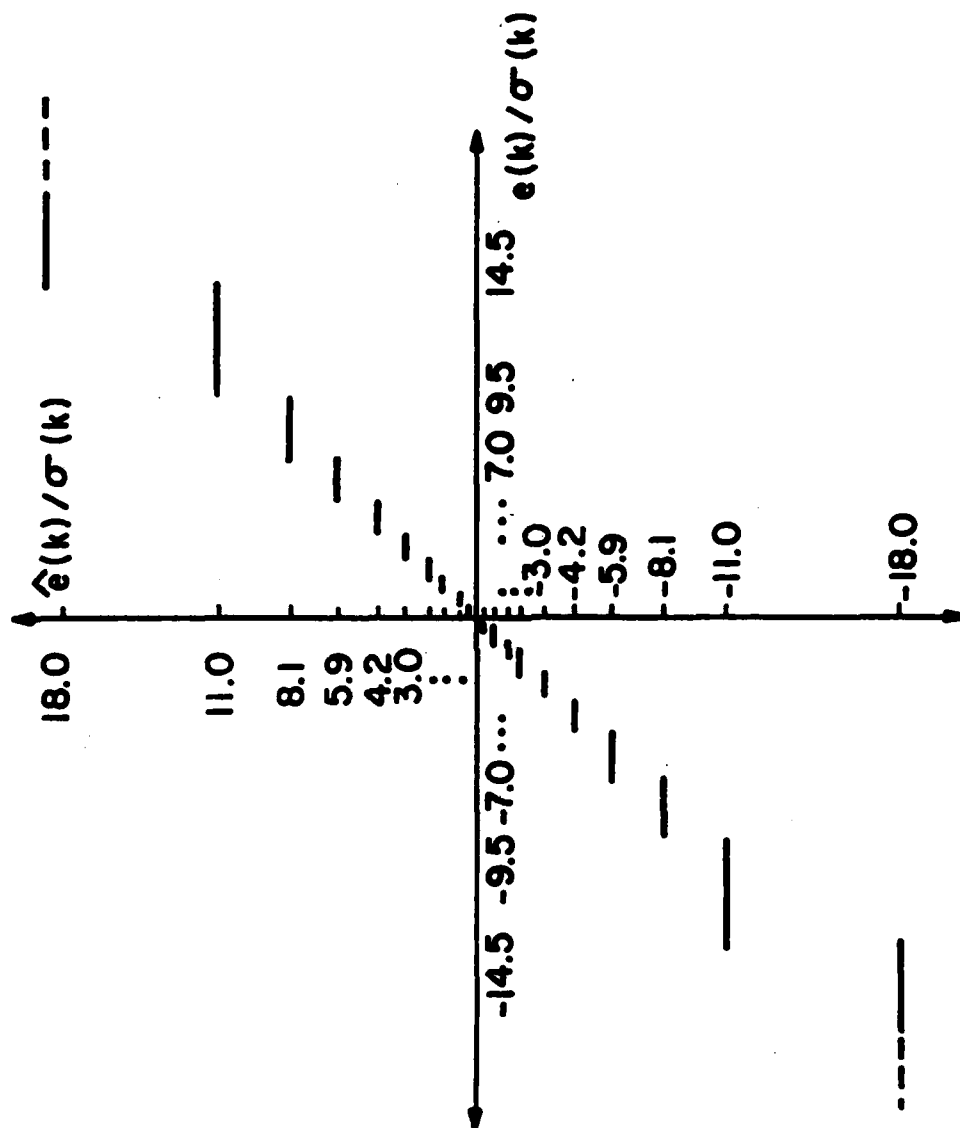
The hand-picked data set used in the comparison is depicted in Fig. 4.2.2 and the resulting logarithmically distributed scaling factors are illustrated in Fig. 4.2.3. Both of these figures incorporate the usual input-output design of past publications. The scaling factors and thresholds agree nearly exactly except for one level in the middle of each side. This aberration is momentarily ignored.

The expansion factors are the parameters that remain unassigned. The expansion factors are used to force the normalized error to the middle of the quantizer. The approximation to the standard deviation should, therefore, remain unchanged if the center of either half of the quantizer is engaged. For this reason and due to the availability of a large number of factors, the middle three factors are set to unity. The remaining upper and lower values are distributed linearly between the largest and smallest expansion factors, respectively. An Aitken double sweep multivariate search [Luenberger, 1965] determined the local maximum of the SNR; pinpointing the maximizing maximum and minimum expansion values.



HAND PICKED SCALING FACTORS AND THRESHOLDS

Figure 4.2.2



LOGARITHMICALLY DISTRIBUTED SCALING FACTORS AND THRESHOLDS

Figure 4.2.3

A table listing of the quantizer thresholds, scaling factors, and expansion factors for speech at a 6400 Hz and a 8000 Hz sampling rate is presented in Fig. 4.2.4. The large number of expansion factors allowed a tighter tracking of the normalized error sequence's movements. This reduced the occupancy frequency in the outer levels of the quantizer. The maximum scaling factor was appropriately reduced to improve the use of the quantizer levels.

#### 4.3 Bit Error Rate

The low channel bit error rate of 0.1% precipitated a noiseless channel approximation. The simulation results, therefore, established a tight upper bound to the system performance in the presence of a 0.1% BER channel. The assumption of no channel errors facilitated the simulation considerably.

An outcome of this assumption is the inclusion of the coding and decoding blocks in the channel model. Since no errors are generated, the estimated quantizer levels  $\hat{q}(k)$ , of the receiver, are exact replicas of the original quantizer levels,  $q(k)$ . There is no reason, therefore, to model the coding and decoding blocks. They are deemed nonessential and removed from the transmitter and receiver. These blocks are, of course, essential to a hardware system and for this reason, they are considered as part of the channel.

This concept is illustrated in Fig. 4.3.1. There, the bold lines denote the simulation flow of information. The dotted lines illustrate the imagined information path through the noiseless channel.

1	T1 (6.4)	T1 (8)	SF <sub>1</sub> (6.4)	SF <sub>1</sub> (8)	EXPN <sub>1</sub>	EXPN <sub>1</sub>
1	.1	.16	0.0	0.00	.5	.50
2,3	.3	.54	.2	.33	.6	.75
4,5	.5	1.1	.4	.76	.7	1.00
6,7	.7	1.7	.6	1.34	.8	1.00
8,9	1.0	2.6	.8	2.11	.8	1.00
10,11	1.3	3.8	1.1	3.13	.9	2.00
12,13	1.7	6.2	1.5	4.48	1.0	3.00
14,15	2.1		1.9	8.00	1.0	5.00
16,17	2.7		2.4		1.0	
18,19	3.3		2.9		1.6	
20,21	4.0		3.6		2.3	
22,23	4.8		4.4		2.9	
24,25	5.8		5.2		3.5	
26,27	6.9		6.3		4.1	
28,29	9.2		7.5		4.8	
30,31			11.0		6.0	

**Figure 4.2.4 Quantizer Thresholds, Scaling Factors, and Expansion Factors for Speech at 6400 Hz and 8000 Hz Sampling.**

The removal of the buffers in the transmitter and receiver is assumed when this modification is made. The buffers are removed due to a combination of the fixed length code condition and the temporary assumption of no channel errors. The removal of the buffer takes with it the buffer control signal. Since there is no error protection or correction coding needed and further since a fixed length code is used, transmission need not be buffered. The original system incorporated a variable threshold quantizer that adapted according to a buffer control signal. This is now unnecessary.

The modifications noted in this chapter are justified by a side-by-side comparison of processed and unprocessed speech sentences and the predictor's computed performance. At this point, the SNR of the predictor was recorded in excess of 22 dB for sentence 1 and over 27 dB for sentence 6. A side-by-side listening session established the speech estimate as being a very good representation of the original speech.

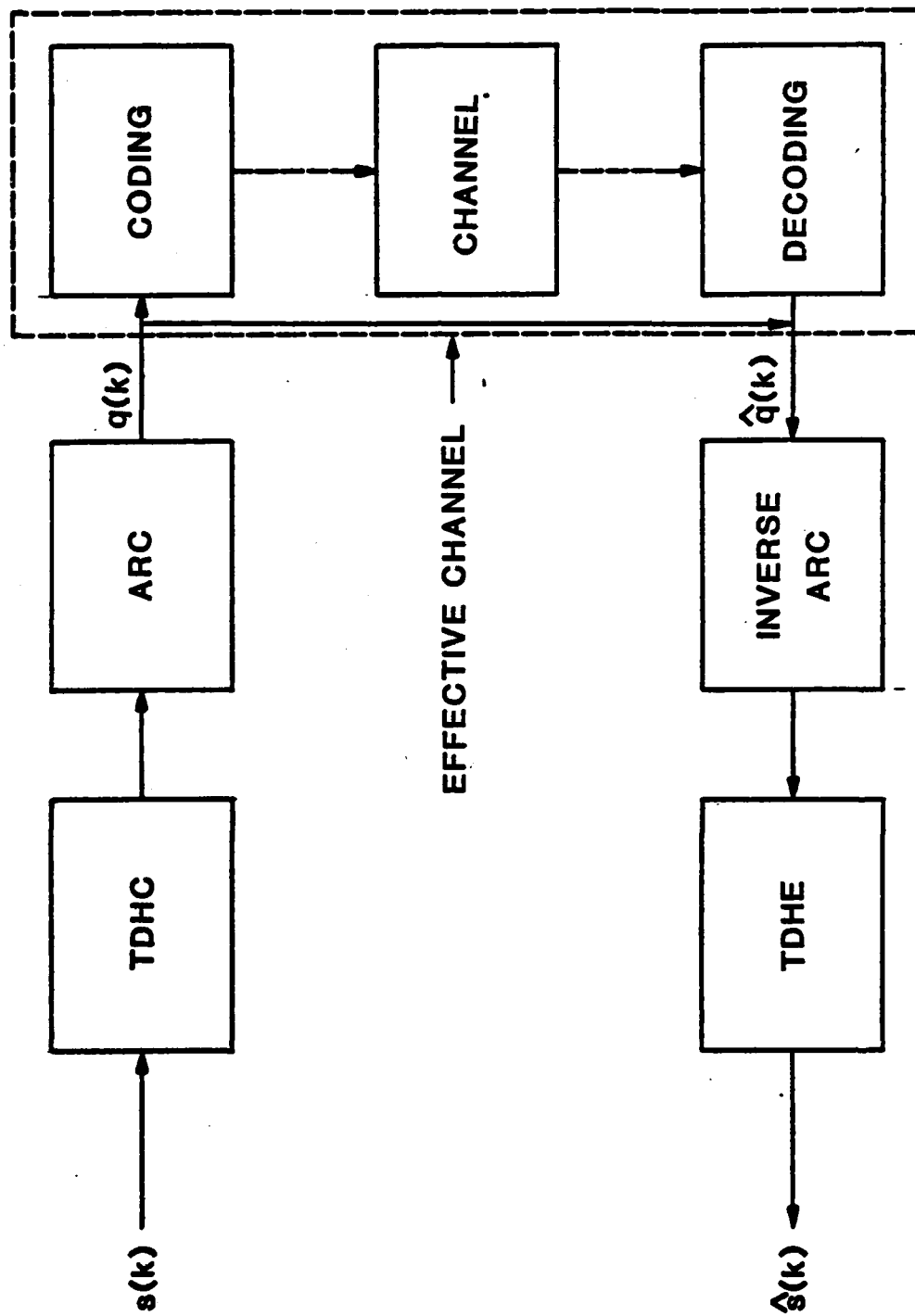


Figure 4.3.1 Channel with Temporary BER Assumption

## CHAPTER V

### HARDWARE INDUCED MODIFICATIONS

#### 5.1 Introduction

The refinements that are principally induced by hardware considerations are presented in this chapter. It is subdivided into four sections. They are 1) serial pitch picking, 2) 8 kHz sampling rate, 3) hardware compatible ARC, and 4) finite word length. The section titles refer to the hardware desired condition which instigated the investigation into that portion of the algorithm.

#### 5.2 Serial Pitch Picking

The original algorithm chose pitch by first buffering a block of speech samples. The first and last third of the samples in the block are then scanned for the maximum speech sample magnitude and the clipping level is set to a specified percentage of this maximum value.

$$C_K = \gamma \cdot \max\{S(n) : n \in [0, \frac{L}{3}] \text{ or } n \in [\frac{2}{3}L, L]\}$$

where  $L$  is the block length in samples,  $C_K$  is the clipping level for the  $k^{\text{th}}$  determination of pitch, and  $s(n)$ ,  $n \in [0, L]$  are the speech samples in the buffer. Another buffer of the same length,  $L$ , is filled by three value center clipping the buffered speech samples. A correlation computation is performed on the clipped speech and a shifted version of it. The correlation is computed over the specified search range for each shift between its maximum and minimum shifts. The shift value associated with the maximum correlation computation is accepted as the pitch period. A window is formed using this value and compression or

expansion is performed. This technique of pitch detection is especially attractive due to its simple hardware implementation.

After the transmitter compresses the speech, the speech buffer is shifted by two pitch periods and refilled with new samples. By shifting out two pitch periods of processed speech and shifting in two pitch periods of new speech, the next clipping level,  $C_{k+1}$ , can be different than the previous clipping level,  $C_k$ . This is caused by the term "max{.}" in the clipping level computation, which can result in different values for the two speech sample buffers. Since the entire buffer of speech is clipped using a single clipping level, the unprocessed samples of the previous speech buffer must be reclipped with the new clipping level. This means that the entire previously clipped speech is useless and must be discarded. Furthermore, clipping cannot take place until the speech buffer is completely filled and a new clipping level is determined.

The problem of waiting for the speech buffer to fill before starting the clipping process and the problem of clipping speech more than once before processing the associated speech samples are eliminated by serial pitch picking [Arora, 1983]. The solution to this time wasting problem is termed serial pitch picking. The serial technique uses a scaled approximation to the speech signal energy as a sample by sample updated clipping level. The clipping level,  $C_k$ , is

$$C_k = \beta \cdot E_k$$

where

$$E_k = \alpha \cdot E_{k-1} + (1-\alpha) |S_k|.$$

$\beta$  is the scaling constant and  $\alpha$  is a constant related to the length of

the transient response. The response of this computation is described by

$$H(Z) = (1-\alpha) \frac{Z}{Z-\alpha}$$

which describes a low pass filter whose time constant, in samples, is a function of  $\alpha$ . The time constant,  $n$ , in samples is

$$n \approx \frac{1}{1-\alpha} .$$

As the new speech enters the buffer, the energy is updated, the clipping level is determined, and the speech is three value center clipped using this value. The clipped speech sample is shifted into the clipped speech buffer and the routine is ready to accept the next speech sample. If the previously detected pitch period is large, a large number of samples must be accepted and clipped to set up the buffers for the next pitch detection. The large number of new samples will take a proportionately large hardware time period to clip. This will not present a problem since the time between pitch detections is also proportional to the pitch period length. (As the number of samples involved in the compression increases, the time required to compress these samples increases and the frequency of pitch detection decreases.)

The unprocessed speech of the previous speech buffer was clipped at the correct clipping levels. The clipped speech associated with the unprocessed speech is, therefore, retained and the redundancy of the clipping in the block pitch picking is eliminated. The clipped speech is generated as the samples enter the speech buffer. Therefore, the

problem of waiting for the speech buffer to fill found in block pitch picking is also eliminated.

The computational simplicity of the three value center clipped speech correlator is sustained. Therefore, serial pitch picking retains the simple hardware implementation of the block pitch picking and yet eliminates the problems associated with it. The comparison of the two techniques is illustrated in Fig. 5.2.1. The difference in the algorithm for the receiver is that at most one maximum pitch period of samples could be used whereas the transmitter's maximum is twice this value.

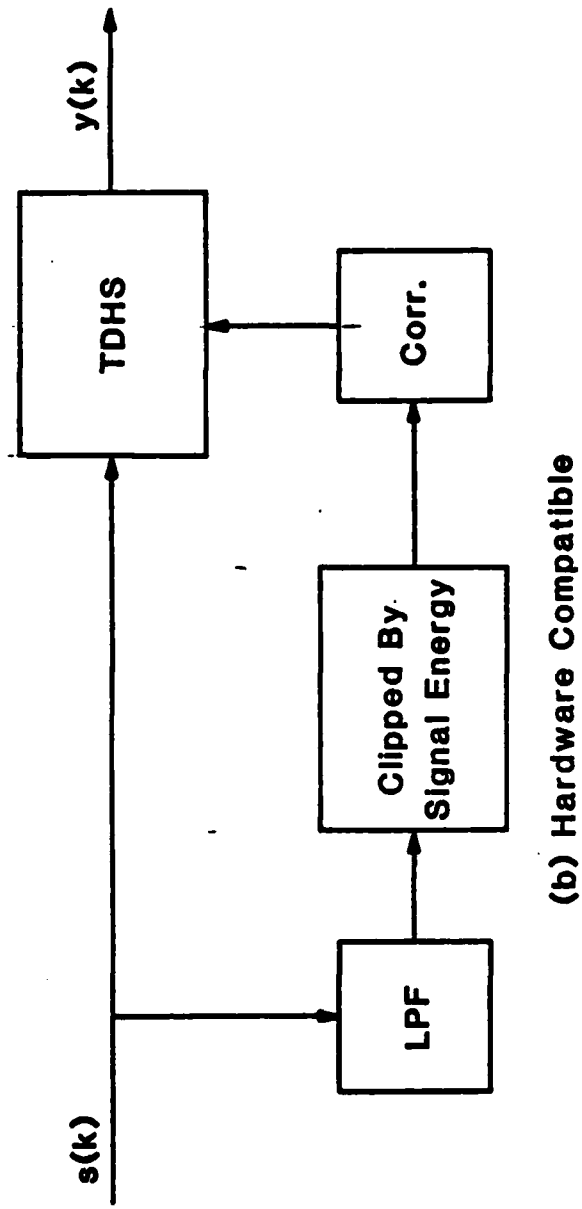
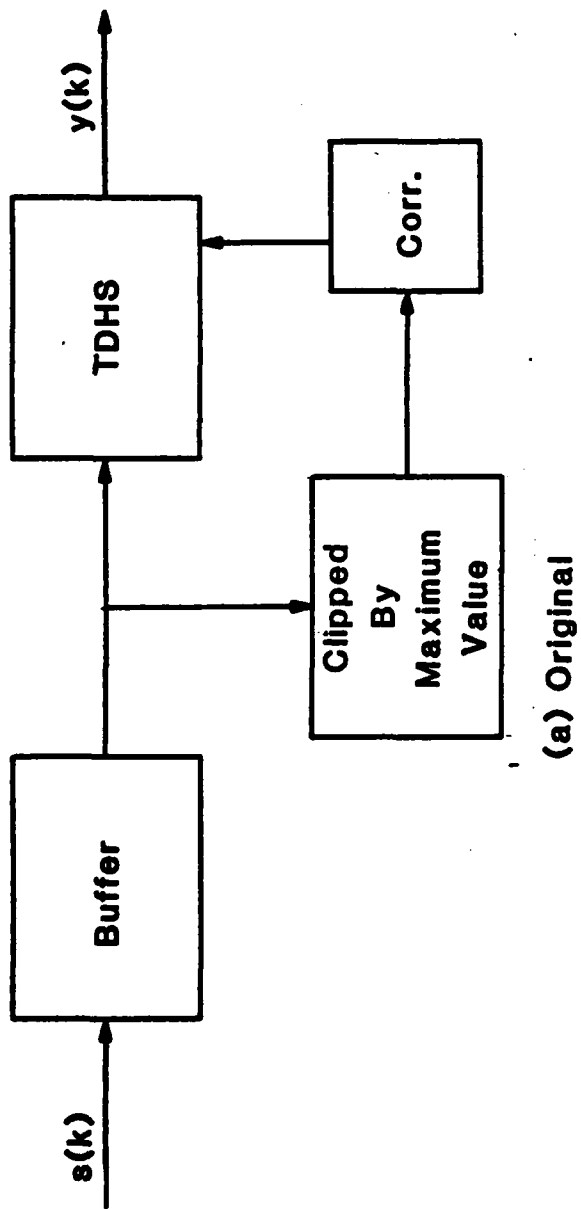
Numerous tests were performed and the suggested parameter values are

$$\beta = 1.5$$

$$\alpha = 0.992$$

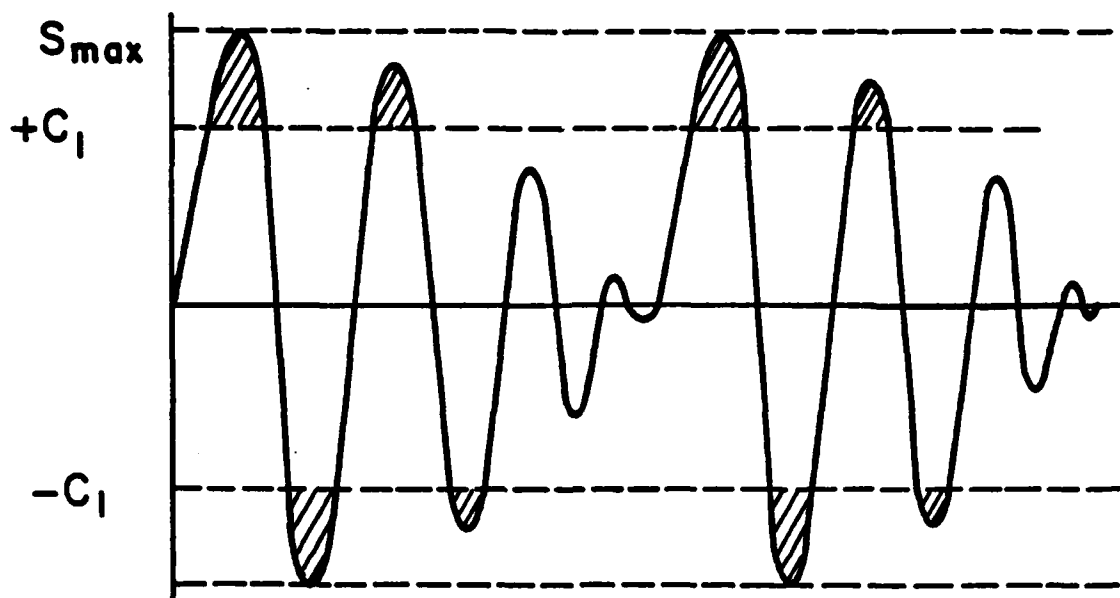
This  $\alpha$  value allowed the proper tracking of the signal energy required by the technique and the above  $\beta$  value insures the success of the clipping routine by properly scaling the resultant signal energy. The fundamental concept of three value center clipping is to retain only the first couple of positive and negative going pulses of each pitch period [Schafer, 1976]. Figure 5.2.2(a) shows how block pitch extraction succeeds in clipping in this manner. Figure 5.2.2(b) demonstrates serial clipping's similar performance and Fig. 5.2.2(c) is the independent result of the two methods.

Two problems developed in the serial clipping implementation. Half pitch was sporadically picked in the voiced segments and a buzzing sound appeared in the output speech during unvoiced segments. This buzzing sound is attributed to the repeated minimum pitch periods picked in

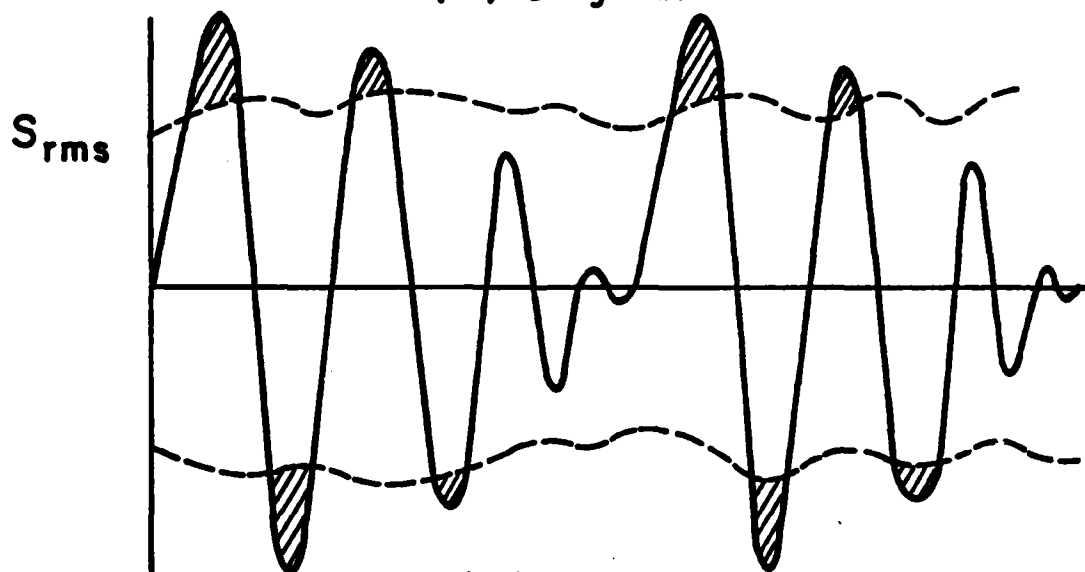


## TDHS ALGORITHMS

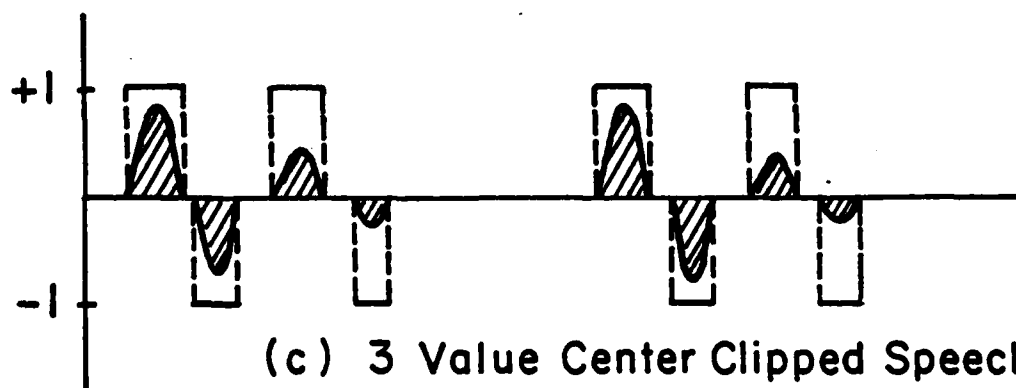
Figure 5.2.1



(a) Original



(b) Final



(c) 3 Value Center Clipped Speech

## CENTER CLIPPING ALGORITHMS

Figure 5.2.2

unvoiced segments by the serial pitch extraction technique. Both problems are charged to the clipped speech sequence, which alone, affects the correlation computation.

The half-pitch problem is solved by implementing a digital low pass filter and by modifying the processing of the correlation computations. The actual correlation computation, itself, remained unaltered. However, the processing of the elements of the correlation vector was changed. The routine now seeks the largest of the correlation vector's one sample local maxima. A one sample local maximum is defined as an element whose magnitude is greater than or equal to its immediate neighbors. The largest local maxima indicates the pitch period as did the vector's maximum element in the original algorithm. The incorporation of this strategy eliminated the possibility of choosing either the maximum or minimum pitch when they represent the rising or falling side, respectively, of a correlation sequence's local maximum. This technique aided in picking true pitch in the voiced regions.

The half-pitch problem is further mastered by installing a low pass filter between the speech sequence and the clipper. It is a 9th order linear phase FIR filter (see Appendix D). The filter is designed to smooth out the anomalies of the individual voiced segments, thereby forcing the pitch periods to appear more similar. This aids the correlator by coercing the clipped speech of each pitch period to be more alike. The result of these two methods is the elimination of half-pitch picked in voiced segments.

The problem of repeated minimum pitch period picked in the unvoiced segments is corrected by implementing a controlled random pitch

generator. Whenever the minimum pitch period is picked, a pseudo-random pitch is chosen and employed as the correct pitch. The random pitch is restricted to the lower half of the search range so that a large randomly picked pitch period in an unvoiced region will not adversely affect the beginning of a voiced segment. Since the pitch in the voiced regions is considerably larger than the minimum value, the random pitch picking should not be engaged and no effect to the voiced regions should occur. If a minimum pitch period is picked in a voiced region, it is an incorrect pitch and randomly choosing another pitch period will not harm and may improve the system performance. This variation to the pitch picking algorithm eliminates repeated minimum pitch estimations in the unvoiced regions. The problem is, therefore, solved.

For the first time, the receiver pitch extraction algorithm differed from that of the transmitter. It is noted that a forward searching correlator in the receiver has less pitch information available to it than did its counterpart in the transmitter. The larger probability of incorrectly choosing pitch at the boundaries of voiced regions results in serious distortions of the output speech. Therefore, whenever pitch is present in the compressed signal estimate  $\hat{y}(k)$ , it is critical that the information is extracted and used. The correlator has been modified to include a larger searching range. The beginning of this search range is located in previously expanded compressed speech. The buffer sizes in the receiver must, therefore, be larger since this previously processed speech is normally discarded. This variation led to improved performance of the previously unsatisfactory receiver serial pitch extraction algorithm. Further testing into the possibilities of reducing the search

range from 20 to 100 to a digital hardware compatible range such as 24 to 88 was performed. Multiple testing on the boundaries of the searching range of the correlator proved that no perceptable difference in speech quality accompanied the reduction of the searching range as far as 24 to 88. This searching range is, therefore, adopted in the final algorithm.

### 5.3 8 kHz Sampling Rate

Throughout the term of this research, studies paralleling those described for a system operating on a 6400 Hz sampled speech signal were performed on an alternate system designated for a 8000 Hz sampling rate. The 8 kHz sampling rate is a standard industry rate and repeatedly presented itself as a possible alternative to the 6400 Hz system. Eventually, the superior availability of reliable 8 kHz filters proved to be the convincing argument for the shift to an 8 kHz system.

The modifications necessary included a reduction in the number of quantization levels to  $2^4 - 1 = 15$ , instead of the 31 levels previously used. The 31 levels of the 6.4 kHz system produced such fine quantization that a fixed (first order) linear predictor was considered. The quality of the fixed predictor system at 6.4 kHz is identical to that of the adaptive predictor at 6.4 kHz. The reduction in available quantization levels and the goal to produce equivalent quality speech at an 8 kHz sampling rate, dictated the use of an adaptive predictor. The thresholds, scaling factors, and expansion factors are all selected using the technique that generated the 31 level values presented in Chapter IV.

Other parameters that require adjustment are the alpha used in the signal energy, the block length and searching range of the correlator, and the LPF coefficients. The alpha value reported in the previous section is the corrected value for the 8 kHz system. The searching range limits are changed to 40 to 110, which reflects nearly an  $8000/6400 = 5/4$  scaling factor. The 8 kHz filter with 1000 Hz cutoff frequency presented in the Appendix is the filter used in the 8 kHz sampling rate system.

#### 5.4 Hardware Compatible ARC

The main processing integrated circuits of the proposed hardware implementation are the INTEL 8085 and NEC 7720. Although each in itself is a very versatile chip, neither supports division capabilities. This hardware consideration leads to significant modifications in the adaptive residual coder. Specifically, the adaptation of predictor coefficients and the normalization of the input to the quantizer use a division operation which must be eliminated.

The original system adapted the coefficients according to the steepest descent of  $e^2(k)$  [Melsa and Cohn, 1975].

$$a_1(k+1) = \delta b_1 + (1-\delta) \left[ a_1(k) + \frac{\hat{g}\hat{y}(k-1)e(k)}{\langle |\hat{y}(k)| \rangle^2} \right]$$

where  $\langle |\hat{y}(k)| \rangle$  is a biased exponential time average of  $|\hat{y}(k)|$  [Melsa and Pande, 1981].

The division by  $\langle |\hat{y}(k)| \rangle^2$  must be modified. This is accomplished by adopting a new adaptation procedure. The final algorithm uses a

fixed gain (percentage) adaptation.

$$a_1(k) = b_1 + a_1^*(k)$$

$$a_1^*(k+1) = \beta(a_1^*(k) + \text{sgn}\{\hat{e}(k), \hat{y}(k-1)\} \max\{g|a_1^*(k)|, a_{\text{bias}}\})$$

$$a_1(k+1) = b_1 + a_1^*(k+1)$$

where the recommended values are

$$b_1 = 1.79$$

$$b_2 = -1.40$$

$$b_3 = 0.57$$

$$b_4 = -0.15$$

$$\beta = 0.95$$

$$a_{\text{bias}} = 0.001$$

$$g = 0.15$$

The  $b_i$  are determined using the technique suggested by R.A. McDonald (1966). The term,  $\max\{\cdot\}$ , computes the biased percent change desired. The  $a_{\text{bias}}$  term helps the coefficient move from the zero level. Without such a term, a coefficient which attained a zero value would remain at zero. The term,  $\text{sgn}\{\cdot\}$ , is a first order estimate of the direction in which the coefficient should move to improve the predictor performance.

The original quantizer accepted the error sequence from the predictor and normalized these values. The normalized values were then compared to the thresholds and mapped to appropriate quantization levels,

$$e(k)_{\text{NORM}} = e(k)/\sigma(k)$$

$$e(k)_{\text{NORM}} \propto T(1) = SF(1)$$

The division by the approximation to the standard deviation,  $\sigma(k)$ , must be eliminated. This division is removed by comparing the error sequence,  $e(k)$ , to adaptive thresholds,

$$e(k) \propto T(i,k) - SF(i,k)$$

$$T(i,k) = \sigma(k) \times T(i)$$

$$SF(i,k) = \sigma(k) \times SF(i)$$

These modifications resulted in no noticeable difference in output speech quality.

### 5.5 Finite Word Length

With any real time system a restriction on the word length is imposed. The integrated circuitry proposed for the construction of a real time system uses 16 bit words. Therefore, a software simulation which restricts all wordlengths to a maximum of 16 bits is desired. It should be noted that it is possible to increase the word accuracy to some integer multiple of 16, such as 32. This requires a much more complicated hardware algorithm since each software computation would in fact require more than one hardware computation. Keeping track of the order and location of multiple word computations further complicates such a scheme. Therefore, a 16-bit finite word length simulation is highly desired.

In fact, a 16-bit finite wordlength simulation resulted in very good quality output speech as compared to the output of the simulation having no such restriction. In all computations and representations the most significant bit is left unoccupied to give a factor of 2 margin of

error. This is done to virtually eliminate the possibility of severely distorting overflows. Reiterating, the performance of the finite wordlength simulation is comparable to that of the final unrestricted wordlength simulation. A 16-bit wordlength simulation is, for this reason, adopted.

The speech quality as perceived by side-by-side comparisons of a taped output version of the speech and input was qualified as very good. The predictor SNR is in excess of 23 dB for sentence 1.

## CHAPTER VI

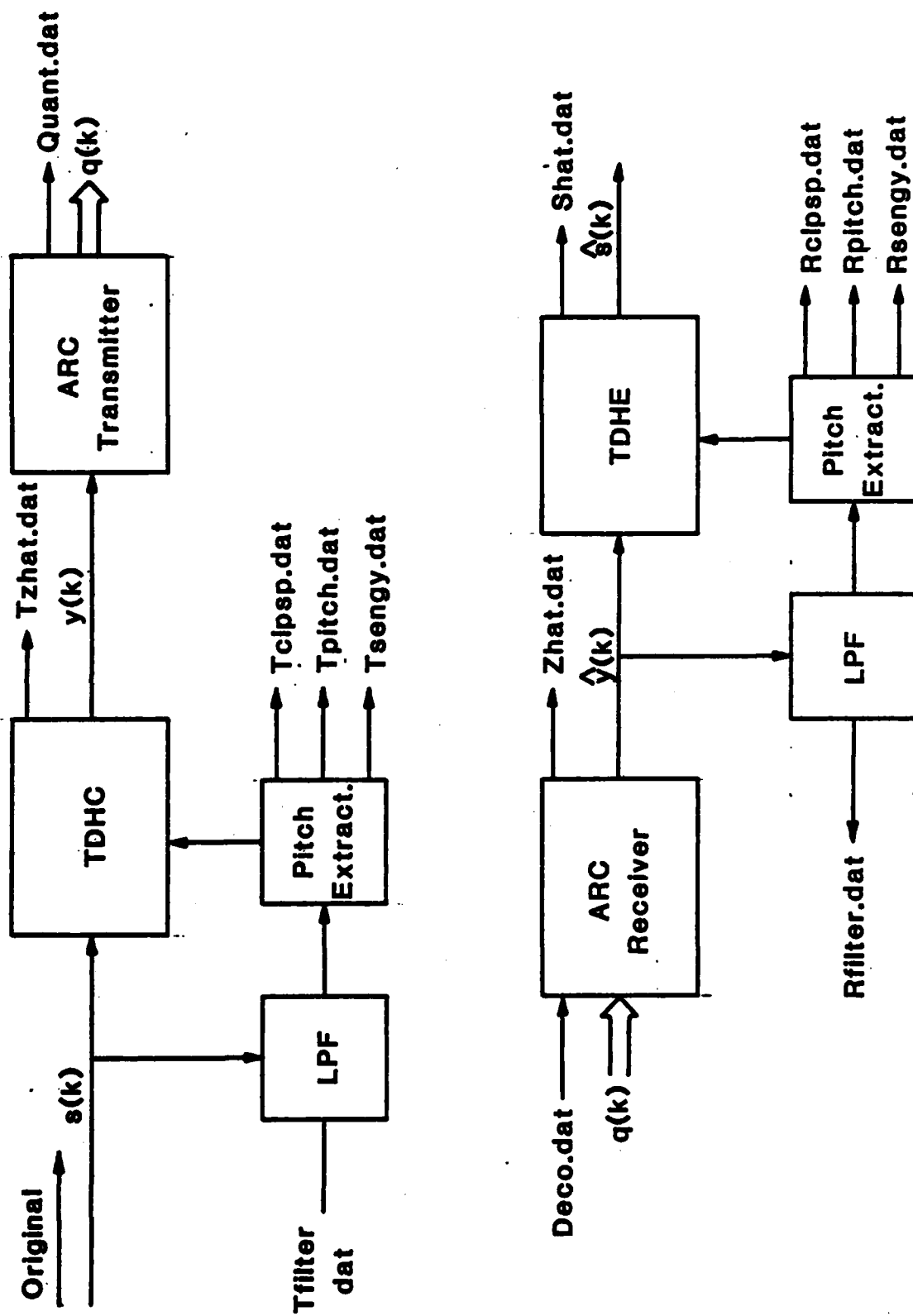
### METHODOLOGY

#### 6.1 Introduction

This chapter introduces the software capabilities and describes the associated output files of the simulation in the context of their usefulness through past research activities and for the benefit of future researchers using this software. The output files are presented first to simplify the task of explaining the program options which employ these files. They are illustrated as output files in Fig. 6.1.1. The file names will precede their individual discussion and a (T)(R) at the beginning of a file name indicates that an option exists to output this file from the transmitter and/or the receiver. If it is generated in the transmitter, the first letter of the name is a T, and if the receiver is the source, the file name begins with an R.

#### 6.2 Files and Options

The changes incorporated in the pitch extraction algorithms resulted in the generation of a number of files which aided in the evaluation of the technique under consideration. Those associated with this topic are (T)(R) PITCH.dat, (T)(R) CLPSP.dat, (T)(R) SENGY.dat and (T)(R) FILTER.dat. (T)(R) PITCH.dat proved to be a vital file in the investigation of pitch extraction. It contains the values of pitch picked in the respective algorithm, transmitter or receiver. A simple listing of these files contains a large quantity of information. With this listing, it is possible to make a quick determination of whether the pitch is being extracted properly or not. Given a sampling rate of 8000 Hz,



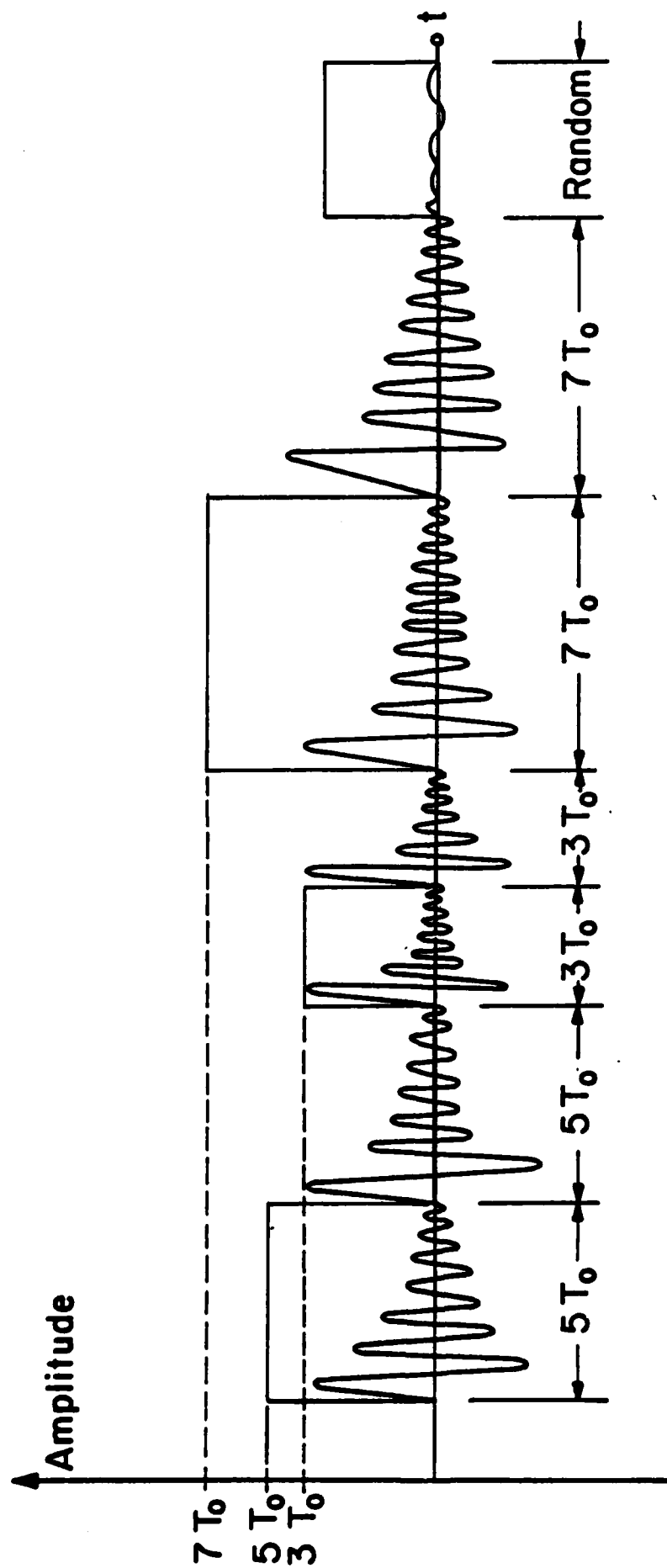
Final System Algorithm and Output Files

Figure 6.1.1

the expected pitch period length in voiced male segments is approximately 60 samples. The algorithm's agreement with this can be expeditiously evaluated from the listing. A critical condition of the pitch extraction algorithm is that it should choose the same pitch period in the receiver as it chose in the transmitter. Side-by-side comparison of the files will determine if this condition is fulfilled.

PITCH.dat illustrated the periodicity of pitch picked in silence regions by the serial pitch extraction algorithm and precipitated the implementation of the random pitch picker. PITCH.dat forms the input of the program titled PICHMAK.ftn which transforms the pitch periods into a series of rectangular pulses of height equal to the pitch period and duration equal to twice the pitch period. The output of this file is compatible input for the plotting routine used throughout the research called DATAPLOT.ftn. The usefulness of the hard copy output is that it can be laid upon a speech segment giving a pictorial account of the pitch extraction algorithm's performance. This is illustrated in Fig. 6.2.1. Although this is a more precise and reliable report of the effectiveness of the pitch extraction routine, the cost, in time, required to spool the results to the plotter, necessitates the use of the pitch period lists in multiple run testing.

(T)(R)CLPSP.dat is a list of the three value center clipped speech. It is also compatible with the plotting program, the output of which determined the potency of the serial clipping routine. The expected output of such a clipping routine, with speech as an input, is a series of positive one's occurring at the large positive going section of the pitch pulses and negative one's for their negative going counter parts.



# GRAPHICAL PITCH EXTRACTION

Figure 6.2.1

Zeros are expected to fill in the gaps. Laying this on top of the input speech file is, again, effective in ascertaining the clipping action. The time associated with obtaining the hard copy should be evaluated before performing a multiple run evaluation. This file illustrated the proximal relationship between the block pitch extraction's clipping routine and that of the serial pitch extractor.

(T)(R)SENGY.dat is a listing of filtered speech signal energy (compressed speech, in the receiver). Used with the plotting program, this file demonstrated the effectiveness of the signal energy computation. It is necessary to compare the signal energy plot to the output of the filter and not to the original speech, since clipping will be performed on the filtered speech. Comparison of the signal energy to the filtered speech assisted in calibrating the clipping value, CLPP. CLPP is the scaling value of the signal energy; the product being the instantaneous clipping level.

(T)(R)FILTER.dat is that file whose hard copy is compared to the signal energy computation as discussed in the previous paragraph. It is the low pass filter output of the pitch extraction units. It's plot verified the filter's smoothing effect on the input speech.

The combination of these files depicted the performance of each component of the pitch extraction unit. Their aid was unparalleled in the modification of the TDHS algorithms and it is suggested that they continue to be used in the evaluation of any further simulation adjustments.

TZHAT.dat. (simply, ZHAT.dat. in the transmitter) contain the compressed speech files. A side-by-side listening session comparing a

taped version of these files, as well as, the examination of their hard copy plots greatly contributed in determining the potency of the TDHS units. The hard copy is expected to be a plot highly related to the original, whose time axis is scaled according to the compression ratio adopted. The taped version should reflect this same time axis scaling.

QUANT.dat and DECO.dat are the input and output of the channel, respectively. They are sequences of quantization levels and are equivalent when the noiseless channel approximation is engaged. When a noisy channel is employed, DECO.dat is the error infected version of QUANT.dat.

TSTATS.dat is a file output by the transmitter only. It contains a presentation of the system performance criteria employed in the original system. TSTATS.dat includes items such as the average signal-to-noise ratio of the predictor and the parameters used in the quantizer. A sample of this file is contained in Appendix E. Since the different SNR measurements pertain to the transmitter unit, this file was primarily commissioned in the ARC evaluations.

SHAT.dat contains the simulation's estimate of the original speech file. The taped reproduction of this formed the most highly weighted estimation of the system's effectiveness. Frequently, the quality of the output file, as perceived by listeners, was the only criteria used in decisions. This file, therefore, formed the most important measure of the system's performance.

The program options are illustrated in Fig. 6.2.2. There are four major options available to the user. The simulations available are an algorithm that exercises the TDHS unit only (without the ARC) and one that concatenates the TDHS and ARC blocks. Each of these options has the

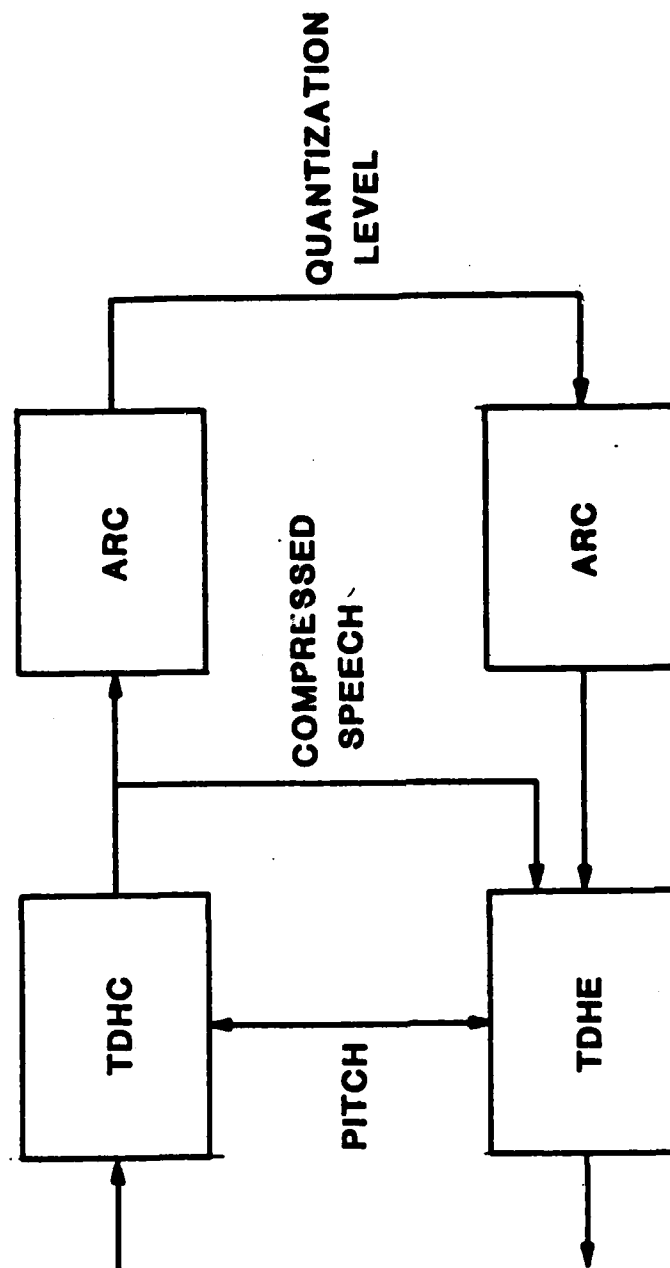


Figure 6.2.2 Algorithm Capabilities

option of passing the pitch picked in the transmitter, as side information, to the receiver. No cost to channel capacity was evaluated for the separate options with the exception of the option of TDHS-ARC with no pitch passed. This option represents the final algorithm. The separate options assisted in assessing the noise due to the TDHS algorithm, the ARC blocks, and the incorrect pitch in the receiver. The comparisons of the different options demonstrated the magnitude of each of these noise sources.

The source files for the unrestricted word length simulations are presented in Appendix A. Reference should be made to these when deciphering Appendix B, the finite word length simulations' source files. Both sets of source files are driven and fused by the included indirect command file. Appendix C contains the source files of supporting programs which aided in the development process.

## CHAPTER VII

### CONCLUSIONS

The final system incorporates all of the specifications of the Chapter I and forms a valid simulation of a finite word length time domain harmonic scaling-adaptive residual coder system. The final predictor SNR is in excess of 22 dB for the sentence

"Cats and dogs each hate the other."

The critical topics for those who are familiar with TDHS-ARC system are contained in Chapters III, IV, and V.

The issue herein reported still maintains questions unanswered and forms possible topics of future investigation. The optimization of the technique used to distribute scaling factors, thresholds, and expansion factors is one example. It seems that the maxima of these sets of parameters could be reduced to the extent of precipitating added accuracy in the finite word length simulations. An investigation into the removal of the random pitch generator is pressed by the effectiveness of the modified correlator. Random pitch is picked few times in the transmitter, usually at the beginning or end of the sentence and rarely is random pitch picked successively. The receiver results are similar. For the above sentence, the receiver picked successive random pitch only once. It does not seem to be worthy of the added software complexity to implement this technique. Investigations into the difference of parameter values of the receiver as compared to the transmitter need to be accomplished (such as, the time constant of the clipping signal energy computation).

The final algorithm incorporates the suggested TDHS-ARC structure with modifications to the original generated by the elements of the original proposal and by hardware restrictions. For the TDHS modifications, emphasis was placed on finding a hardware compatible pitch detector. The new adaptation scheme is the major variation found in the ARC. The final hardware simulations resulted in good quality speech in the presence of up to 0.1% BER, ascertained by side-by-side comparison of a taped reproduction.

The goal, to design a simulation by means of algorithm refinements that is easy to implement in hardware and yet does not degrade the present performance, has been met.

## APPENDIX A

### FINITE WORD LENGTH SIMULATION SOURCE FILES

The necessary modules to run a simulation are:

- 1) CMD16.CMD (Indirect command file)
- 2) PFWL.FTN (Parameter builder)
- 3) TFWL.FTN (Transmitter)
- 4) C8K.FTN (Channel)
- 5) RFWL.FTN (Receiver)

The task transmitter and receiver modules should be built from the source files using the following switches:

> UNITS = 15

> ACTFILE = 15

All other source files use the default values. A specific file management procedure is dictated by the indirect command file. This can easily be modified to fit the specific needs of a multiple run use.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	5
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	---

1) P16.TSK

2) T16.TSK

3) R16.TSK

#### 4) PICHMAK.TSK

**5) COK.TSK (REQUIRES OPERATOR MONITOR)**

## AND INPUT

6) E.G., S11689.DAT

**NOTE THAT EARLIER VERSIONS OF THE TRANSMITTER AND RECEIVER ARE NOT COMPATIBLE WITH THIS COMMAND FILE SINCE THEY REQUIRE THAT THE OPERATOR MONITORS THEM FOR INTERACTIVE DATA.**

**.ENABLE SUBSTITUTION**  
**.ENABLE LOWERCASE**

**.SETN CIN 8**

SETN S2N 8

**SETS.**

**.SETN NUMFIL &**

**.SETS NUM '123456789'**

**.ASK TOUT DO YOU WANT**

**.IFT TOUT .SETN SIN 1**

**.ASK ROUT DO YOU WANT**

..IFT ROUT .SETN S2N 1

**...ASK CHN DO YOU WANT**

IFT CNN .SETN C1N 1

**.INC NUMFIL**

**RUN IS FOR THE TRANSM**  
**RUN D1C**

**RUN P16**



```

      .INC NOMPIL
      ;THIS RUN IS FOR THE TRANSMITTER OPTION FILE
      RUN P16

```

```

.VAIT
PIP TOPT.DAT=PARAM.DAT/RE
.15: .ASK Q1 DO YOU WANT THE RCVR OPT FILE=TRAN OPT FILE
      .IFF Q1 .GOTO 35
      .SETN NEWNUM NUMFIL
      .SETS NEWDAT NUM(NEWNUM:NEWNUM)
      .SETS NEWNAM "ROPT.DAT;" + NEWDAT
      .SETS OLDNAM "TOPT.DAT;" + NEWDAT
      PIP 'NEWNAM' = 'OLDNAM'
      .VAIT PIP
      .GOTO 45
.35: .VAIT
      ;THIS RUN IS FOR THE RECEIVER OPTION FILE
      RUN P16
      .VAIT
      .SETN NEWNUM NUMFIL
      .SETS NEWDAT NUM(NEWNUM:NEWNUM)
      .SETS NEWNAM "ROPT.DAT;" + NEWDAT
      PIP 'NEWNAM' = PARAM.DAT/RE
      .VAIT PIP
      .ASK Q2 DO YOU WANT ANOTHER RUN
      .IFF Q2 .GOTO 75
      .INC NUMFIL
      .ASK Q3 DO YOU WANT THE SAME TOPT FILE AS THE PREVIOUS ONE
      .IFF Q3 .GOTO 65
      .SETN NEWNUM NUMFIL
      .SETN OLDNUM NUMFIL-1
      .SETS NEWDAT NUM(NEWNUM:NEWNUM)
      .SETS OLDDAT NUM(OLDNUM:OLDNUM)
      .SETS NEWNAM "TOPT.DAT;" + NEWDAT
      .SETS OLDNAM "TOPT.DAT;" + OLDDAT
      PIP 'NEWNAM' = 'OLDNAM'
      .VAIT PIP
      .GOTO 15
.65: .VAIT
      ;THIS RUN IS FOR THE TRANSMITTER OPTION FILE
      RUN P16
      .VAIT
      .SETS NEWDAT NUM(NUMFIL:NUMFIL)
      .SETS NEWNAM "TOPT.DAT;" + NEWDAT
      PIP 'NEWNAM' = PARAM.DAT/RE
      .VAIT PIP
      .ASK Q4 DO YOU WANT THE SAME ROPT FILE AS THE PREVIOUS ONE
      .IFF Q4 .GOTO 15
      .SETN OLDNUM NUMFIL-1
      .SETS OLDDAT NUM(OLDNUM:OLDNUM)
      .SETS NEWNAM "ROPT.DAT;" + NEWDAT
      .SETS OLDNAM "ROPT.DAT;" + OLDDAT
      PIP 'NEWNAM' = 'OLDNAM'
      .VAIT PIP
      .GOTO 55
.75: .VAIT

```

```

.88: .SETN ARG 1
      .SETS NDAT NUMARG:ARGJ
      .SETS NAME "TOPT.DAT;" +NDAT
      PIP CMDPARA.DAT="NAME"
      .WAIT PIP
      RUN T16
      .WAIT
      .IF SIN NE 1 .GOTO 100
      PIP PITCH.DAT=TPITCH.DAT
      .WAIT PIP
      RUN PICHMAK
      .WAIT
      PIP PITCH.DAT;*/DE
      .WAIT PIP
      .SETS TPP "TPP.DAT;" +NDAT
      PIP "TPP"=PPULSE.DAT;1/RE
      .100: .WAIT PIP
            .IF CIN NE 1 .GOTO 90
            RUN C8K
      .90: .WAIT
            .IF C1H EQ 1 .GOTO 95
            PIP DECO.DAT=QUANT.DAT;*/RE
            .WAIT PIP
      .95: .WAIT
            .SETS NAME "ROPT.DAT;" +NDAT
            PIP CMDPARA.DAT="NAME"
            .WAIT PIP
            RUN R16
            .WAIT
            PIP DECO.DAT;*,CMDPARA.DAT;*/DE
            .WAIT PIP
            .IF S2N NE 1 .GOTO 120
            PIP PITCH.DAT=RPITCH.DAT
            .WAIT PIP
            RUN PICHMAK
            .WAIT
            .SETS RPP "RPP.DAT;" +NDAT
            PIP "RPP"=PPULSE.DAT;1/RE
            .WAIT PIP
            PIP PITCH.DAT;*/DE
      .120: .WAIT PIP
            .INC ARG
            .IF ARG GT NUMFIL .GOTO 130
            .GOTO 88
      .130: .WAIT 1W.S

```

**PFWL.FTN**

BY: JAMES D. MILLS, 18 APR 1983

**DESCRIPTION: THIS PROGRAM GENERATES A PARAMETER FILE TO BE USED WITH THE FINITE WORD LENGTH SIMULATIONS. IT OUTPUTS A FILE NAMED PARAM.DAT. THE FOLLOWING ARE INTERACTIVELY SELECTED:**

- 55

[illegible]



```

TYPE *,'
TYPE *,'
TYPE *,'
TYPE *,' TYPE--[SELECTION] (SELECTION) (SELECTION)...--UP TO 9 CHOICES'
ACCEPT 788,ANS
FORMAT(9(11,1X))
CHK=8
DO 778 I=1,9
CHK=CHK+ANS(I)
778 CONTINUE
IF(CHK.EQ.8)GO TO 988
ISELEC=1
197 DO 199 I=1,9
IF(ANS(I).EQ.ISELEC)GO TO (281,282,283,284,285,286,287,288
,289).ISELEC
1
199 CONTINUE
GO TO 193
281 TYPE *,' ENTER NEW VALUE OF "ALP"'
ACCEPT *,ALP
GO TO 193
282 TYPE *,' ENTER NEW VALUE OF "ABIAS"'
ACCEPT *,ABIAS
GO TO 193
283 TYPE *,' ENTER NEW VALUE OF "ALPHA"'
ACCEPT *,ALPHA
GO TO 193
284 TYPE *,' ENTER NEW VALUE OF "G"'
ACCEPT *,G
GO TO 193
285 TYPE *,' ENTER NEW VALUE OF "N"'
ACCEPT *,N
GO TO 193
286 TYPE *,' ENTER NEW VALUE OF "RMSMIN"'
ACCEPT *,RMSMIN
GO TO 193
287 TYPE *,' ENTER NEW VALUE OF "SMIN"'
ACCEPT *,SMIN
GO TO 193
288 TYPE *,' ENTER NEW VALUE OF "CLPP"'
ACCEPT *,CLPP
GO TO 193
289 TYPE *,' ENTER NEW VALUES OF "KBLK1 ITMIN ITMAX"'
ACCEPT *,KBLK1,ITMIN,ITMAX
193 ISELEC=ISELEC+1
IF(ISELEC.LE.9)GO TO 197
GO TO 798
988 CONTINUE

```

```

C*****
C*****
C*****
C*****THE FOLLOWING BLOCK SETS UP THE PARAMETERS OF THE
C*****ADAPTIVE RESIDUAL CODER. IT GENERATES THE THRESH-
C*****

```

C\*\*\*\*\*OLDS, SCALING FACTORS, AND EXPANSION FACTORS USING  
C\*\*\*\*\*THE PARAMETERS SPECIFIED BY THE USER.  
C\*\*\*\*\*

```

12 DO 17 I=1,6
   ANS1(I)=B
17 CONTINUE
   DO 18 I=1,6
   IF(ANS1(I).EQ.1)GO TO 3
18 CONTINUE
   GO TO 1B
3 TYPE *, 'ENTER THE MAXIMUM SCALING FACTOR(>B)'
  ACCEPT *,MSF
  IF (MSF.LE.B.)GO TO 3
1B DO 19 I=1,6
   IF(ANS1(I).EQ.2) GO TO 29
19 CONTINUE
   GO TO 2B
29 TYPE *, 'ENTER THE NUMBER OF QUANT LEVELS (5<NUM<1B1) AND ODD'
  ACCEPT *,KQ
  IF (KQ.LT.5.OR.KQ.GT.1B1)GO TO 1B
2B NQ=KQ/2

```

C\*\*\*\*\*

C\*\*\*\*\*DETERMINATION OF THE QUATIZER PARAMETER VALUES  
C\*\*\*\*\*FIRST, THE SCALING FACTORS ARE GENERATED.  
C\*\*\*\*\*

```

OUT(NQQ)=MSF
LMX=(ALOG1B(MSF))
UP=LMX/(FLOAT(NQ))
LFI(1)=B.
DO 4B I=2,NQ
  LFI(I)=LFI(I-1)+UP

```

```

4B CONTINUE
KK=MSF+1
KK=ALOG1B(KK)
LL=LFI(NQ)
KK=KK/LL
BMAX=1B.**KK
DO 214 I=1,6

```

```

  IF(ANS1(I).EQ.3)GO TO 213
214 CONTINUE
  GO TO 42
213 WRITE(5,212)BMAX,B
212 FORMAT(' THE MAX BASE ALLOWED IS ',F5.2,' AND THE CURRENT BASE '
1      ' IS ',F5.2)
  TYPE *, 'ENTER BASE OF LOG FNC USED FOR SCALING FACTORS(>B)'
  ACCEPT *,B
42 IF(B.GT.B.AND.B.LT.BMAX)GO TO 43
  TYPE *, ' THE BASE CHOSEN WAS OUTSIDE THE ALLOWED RANGE. TRY AGAIN!'
  GO TO 213
43 DO 5B I=1,NQ
  M=LFI(I)

```

```

OUT(I)=B**M-1.
55 CONTINUE
DO 60 I=1,NQ
  T(I)=(OUT(I)+OUT(I+1))/2.
60 CONTINUE
DO 61 I=1,6
  IF(ANS1(I).EQ.4)GO TO 62
61 CONTINUE
GO TO 65
C*****DETERMINATION OF EXPANSION FACTORS.
62 TYPE *, ' ENTER THE LOW EXPANSION FACTOR'
ACCEPT *,LOEXPN
65 DO 66 I=1,6
  IF(ANS1(I).EQ.5)GO TO 67
66 CONTINUE
GO TO 75
67 TYPE *, ' ENTER THE HIGH EXPANSION FACTOR'
ACCEPT *,HIEXPN
75 MID=NQ/2+1
  EXPN(MID)=1.
  EXPN(MID+1)=1.
  EXPN(MID-1)=1.
  EXPN(1)=LOEXPN
  EXPN(NQ)=HIEXPN
  HIDEL=(HIEXPN-1.)/(FLOAT(NQ-MID+1))
  LODEL=(1.-LOEXPN)/(FLOAT(MID-2))
  XX=HIDEL+1.
  VV=LODEL+LOEXPN
  UU=MID-2
  VV=MID+2
DO 110 I=2,UU
  EXPN(I)=VV
  VV=VV+LODEL
110 CONTINUE
DO 120 I=VV,NQ
  EXPN(I)=XX
  XX=XX+HIDEL
120 CONTINUE
C*****INTERMEDIATE PRINTOUT OF QUANTIZER PARAMETERS.
TYPE *, 'WHERE SHOULD THE PRINTOUT GO?(TERM=0,PRINT=1)'
ACCEPT *,I
IF (I.EQ.0) VARE=5
IF (I.EQ.1) VARE=6
WRITE(5,100)ALP,ABIAS,ALPHA,G,N,RHSMIN,SMIN,CLPP,KBLK1,ITMIN,
1 ITMAX,ALAD,NBL
IF (I.EQ.1)WRITE(5,160)KQ,NQ,KQ,NQ,MSF,B,LOEXPN,HIEXPN
WRITE(VARE,160)KQ,NQ,KQ,NQ,MSF,B,LOEXPN,HIEXPN
160 FORMAT(// ' THE FOLLOWING IS A LIST OF SCALING FACTORS. '//
1 ' EXPANSION FACTORS, AND THRESHOLDS FOR A .12. LEVEL.'/
2 ' QUANTIZER(.12. ON A SIDE), WITH A MAXIMUM SCALING FACTOR.'/
3 ' =.F7.4. A LOG SCALE WITH A BASE=.F7.4. WAS USED.'/
4 ' THE HIGH AND LOW EXPANSION FACTORS WERE SET AT .F7.4/'

```



```

ACCEPT 535,ANS2
535 FORMAT(3(11.1X))
CHK=#
DO 54# I=1,3
    CHK=CHK+ANS2(I)
54# CONTINUE
    IF(CHK.EQ.#)GO TO 575
    IFILE=ITSF
    ISELEC=1
545 DO 55# I=1,3
        IF(ANS2(I).EQ.ISELEC)GO TO 555
55# CONTINUE
        GO TO 57#
555 IF(IFILE.NE.#)GO TO 56#
    IFILE=1
    GO TO 565
56# IFILE=#
565 IF(ISELEC.EQ.1)ITSF=IFILE
    IF(ISELEC.EQ.2)ICSF=IFILE
    IF(ISELEC.EQ.3)IOSF=IFILE
57# ISELEC=ISELEC+1
    IF(ISELEC.EQ.2)IFILE=ICSF
    IF(ISELEC.EQ.3)IFILE=IOSF
    IF(ISELEC.LE.3)GO TO 545
    GO TO 423
575 CONTINUE
58# DO 585 I=1,8
    ANS3(I)=#
585 CONTINUE
    TYPE *, ' THE FOLLOWING INDICATE WHICH PLOTTABLE FILES WILL BE '
    TYPE *, ' OUTPUT BY THIS COMMAND FILE.'
    TYPE *, ' [#=NOT OUTPUT, 1=WILL BE OUTPUT]'
    TYPE 58#
    TYPE 59#,ITFF
    TYPE 595,ITSE
    TYPE 60#,ITCS
    TYPE 605,ITPF
    TYPE 61#,IRFF
    TYPE 615,IRSE
    TYPE 62#,IRCS
    TYPE 625,IRPF
59# FORMAT('
595 FORMAT('
60# FORMAT('
605 FORMAT('
61# FORMAT('
615 FORMAT('
62# FORMAT('
625 FORMAT('
    TYPE *, ' TO CHANGE ANY OR ALL OF THESE TYPE---'
    TYPE *, ' [SELECTION] (SELECTION) (SELECTION)...--UP TO 8 CHOICES'
ACCEPT 63#,ANS3
1)TRANSMITTER FILTERED SPEECH SET=.,11)
2)TRANSMITTER SIGNAL ENERGY SET=.,11)
3)TRANSMITTER CLIPPED SPEECH SET=.,11)
4)TRANSMITTER PITCH FILE SET=.,11)
5)RECEIVER FILTERED SPEECH SET=.,11)
6)RECEIVER SIGNAL ENERGY SET=.,11)
7)RECEIVER CLIPPED SPEECH SET=.,11)
8)RECEIVER PITCH FILE SET=.,11)

```



```

800 FORMAT(
810 FORMAT(
820 FORMAT(
830 FORMAT(
840 FORMAT(
850 FORMAT(
      TYPE *,
      TYPE *,
      DO 865 I=1,6
        ANS4(I)=0
855 CONTINUE
      ACCEPT 860,ANS4
      860 FORMAT(6(I1,1X))
      CHK=0
      DO 870 I=1,6
        CHK=CHK+ANS4(I)
870 CONTINUE
      IF(CHK.EQ.0)GO TO 940
      ISELEC=1
      875 DO 880 I=1,6
        IF(ANS4(I).EQ.ISELEC)GO TO 890
880 CONTINUE
      GO TO 930
890 IF(ISELEC.EQ.1)TYPE *, ' ENTER NEW VALUE OF INF'
      IF(ISELEC.EQ.1)ACCEPT *,INF
      IF(ISELEC.EQ.2)TYPE *, ' ENTER NEW VALUE OF ICF'
      IF(ISELEC.EQ.2)ACCEPT *,ICF
      IF(ISELEC.EQ.3)TYPE *, ' ENTER NEW VALUE OF IAF'
      IF(ISELEC.EQ.3)ACCEPT *,IAF
      IF(ISELEC.EQ.4)TYPE *, ' ENTER NEW VALUE OF IQ1F'
      IF(ISELEC.EQ.4)ACCEPT *,IQ1F
      IF(ISELEC.EQ.5)TYPE *, ' ENTER NEW VALUE OF IQ2F'
      IF(ISELEC.EQ.5)ACCEPT *,IQ2F
      IF(ISELEC.EQ.6)TYPE *, ' ENTER NEW VALUE OF IQ3F'
      IF(ISELEC.EQ.6)ACCEPT *,IQ3F
      930 ISELEC=ISELEC+1
      IF(ISELEC.LE.6)GO TO 875
      GO TO 790
940 CONTINUE
C*****CHOICE OF FILTERING
C*****
      TYPE *, ' DO YOU WANT TO FILTER THE PITCH SPEECH?'
      TYPE *, ' (YES=1,NO=0)'
      ACCEPT *,NOFLT
      IF(NOFLT.EQ.0)GO TO 780
      TYPE *, ' WHICH FILTER DO YOU WANT?'
      TYPE *, '
      TYPE *, ' 1)8KHZ INPUT SPEECH WITH FILTER CUTOFF FREQ=6000HZ'
      TYPE *, ' 2)8KHZ INPUT SPEECH WITH FILTER CUTOFF FREQ=1KHZ'
      TYPE *, ' 3)6.4KHZ INPUT SPEECH WITH FILTER CUTOFF FREQ=5000HZ'

```

```

TYPE *.
ACCEPT *.1
DO 665 J=1,5
  HF(J)=H(J,1)
665 CONTINUE
GO TO 715
C*****
C*****
C*****
C*****CHOICE OF PROGRAM PATHS
C*****
755 DO 715 J=1,5
  HF(J)=5
715 CONTINUE
TYPE *. WHICH PROGRAM PATH DO YOU WANT TO TAKE?
TYPE *.
TYPE *. 1)TDHS-ARC WITH PITCH PASSED TO THE RCVR'
TYPE *. 2)TDHS-ARC WITH PITCH INDEPENDENTLY GENERATED'
TYPE *. IN THE RCVR.
TYPE *. 3)TDHS ONLY WITH PITCH PASSED TO THE TDHE ALGORITHM'
TYPE *. 4)TDHS ONLY WITH PITCH INDEPENDENTLY GENERATED.
TYPE *. IN THE TDHE ALGORITHM.
ACCEPT *.IPATH
C*****
C*****
C*****CHOICE OF INITIAL PREDICTOR COEFFICIENTS AND GAIN
C*****
845 TYPE *. THE FOLLOWING ARE THE INITIAL PREDICTOR COEFFICIENTS'
TYPE *. AND GAIN FACTOR.
DO 855 I=1,N
  TYPE 815,1,AB(1)
855 CONTINUE
TYPE 825,GAIN
815 FORMAT( AB(.,12,.)=,2X,F8.4)
825 FORMAT( GAIN=,2X,F8.4)
TYPE *. DO YOU WANT TO ENTER NEW VALUES?
TYPE *. YES=1
ACCEPT *.NAI
IF(NAI.EQ.0)GO TO 835
TYPE *. ENTER THE NEW INITIAL PREDICTOR COEFFICIENTS'
ACCEPT *. (AB(1),I=1,N)
TYPE *. ENTER THE GAIN FACTOR'
ACCEPT *.GAIN
GO TO 845
C*****
C*****
C*****OUTPUT BLOCK
C*****
835 WRITE(3,9)(FNAME(J),J=1,16)
WRITE(3,*)INF,ICF,IAF,IOIF,IQ2F,IQ3F
WRITE(3,*)NBL

```

```

WRITE(3,*)CLPP
WRITE(3,*)KBLK1,ITMIN,ITMAX
WRITE(3,*)ALP,ALAD,ABIAS,G,N,RNSMIN,SMIN
WRITE(3,*)KQ
WRITE(3,*)(T(I),I=1,NQQQ)
WRITE(3,*)(OUT(I),I=1,NQQQ)
WRITE(3,*)(EXP(I),I=1,NQQQ)
WRITE(3,*)ALPHA
WRITE(3,*)ITCS,ITPF,ITSE,ITSF,IRCS,IRPF,IRSE,ICSF,IOSF,ITFF,IRFF
WRITE(3,*)NOFLT
WRITE(3,*)(HF(I),I=1,5)
WRITE(3,*)IPATH
WRITE(3,*)(AB(I),I=1,N)
WRITE(3,*)GAIN
CLOSE(UNIT=3)
STOP
END

```

CCCCC

\*\*\*\*\*

# TEVL: ETLN

BY: JAMES D. MILLS, 18 APR 1983

**DESCRIPTIVE TITLE: TRANSMITTER FINITE WORD LENGTH  
SIMULATION**

DESCRIPTION: THIS PROGRAM USES THE OUTPUT PARAMETER FILE OF PFVL.FTN, RENAMED CMDPARA.DAT, TO SET ALL THE OPTIONS OF THE TRANSMITTER. FOR THIS REASON, THE OPERATOR IS NOT PROMPTED FOR ANY INFORMATION DURING A RUN. THIS PROGRAM REPRESENTS THE FINITE WORD LENGTH SIMULATION OF THE TRANSMITTER FOR THE JDHS-ARC ALGORITHM. THE WORD LENGTH HAS BEEN RESTRICTED TO 16 BITS. THE TRANSMISSION RATE IS ASSUMED TO BE 16KB/S.

```

INTEGER HD(40), IPICH(400), FNAME(16), SQ,Q,VARE6
INTEGER*2 CLPSP(512), STAT
INTEGER*4 STRT,BUF5Z,ISF,ITF,IA4,IB4,IAF,ICF
DIMENSION H(400), SQ(16),ISBUF(512),IFSP(512),IVY(16)

```

# DIMENSIONAL REAL ALPHA

COMMON /PRED/G,ALP,ALAD,ABIAS,EP,EV,8(12)  
 ,SPERB1(200),SPERB2(200),SNRB(200),SNRQB(200),N,KQ,NSPSAM  
 ,ISTAT1(48),IVHAT(12),IV(12),IRMSHN,IA(12)  
 COMMON /RNSB/ENGY1,ENGY2,ENGY3,ENGY4,SPER1,SPER2,NBL,IARG,IRMS

```

C
C
      INTEGER HD(40),IPICH(400),FNAME(16),SO,O,VARE6
      INTEGER*2 CLPSP(512),STAT
      INTEGER*4 STRT,BUFSZ,ISF,ITF,IA4,IB4,IAF,ICF
      DIMENSION H(400),SQ(16),ISBUF(512),IFSP(512),IVY(16)
      REAL ALPHA
C*****
C*****
C*****
      COMMON /PRED/G,ALP,ALAD,ABIAS,EP,EV,B(12)
      1      ,SPERB1(200),SPERB2(200),SNRB(200),N,KO,NSPSAM
      2      ,ISTAT1(40),IVHAT(12),IV(12),IRHSMN,IA(12)
      COMMON /RMSB/ENG1,ENG2,ENG3,ENG4,SPER1,SPER2,NBL,IARG,IRMS
      COMMON /CLIPP/ALPHA,ICLPP,ISEC
      COMMON /ADDN/STAT(40)
      COMMON /VARE6/VARE6
      COMMON /IFILE/ITCS,ITPF,ITSE,ITSF,IRCS,IRPF,IRSE,ICSF,IOSF,ITFF,IRFF
      1      ,IPATH
      COMMON /FILT/HF(5),IX(9),IFSC,NOFLT
      COMMON /FVL/ISF,INF,ITF,ICF,IAF,IOIF,IQ2F,IQ3F
      COMMON /IOCNT/IOCNT,INSZ,ILSZ
C*****
C*****
C*****
      MODN(K) = K - (K-1)/16 * 16
C*****
C*****
C*****
C*****
C*****
C*****
      OPEN(UNIT=8,TYPE='OLD',NAME='CMOPARA.DAT') IOPEN THE OPTION FILE.
C*****
C*****
C*****
      OPEN THE SPEECH FILE; READ THE HEADER AND SPEECH.
C*****
      READ(8,16)FNAME
      10  FORMAT(16A2)
      OPEN(UNIT=1,TYPE='OLD',READONLY,NAME=FNAME,SHARED) I OPEN SPEECH FILE
      READ(1,20)INSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
      20  FORMAT(6I5,10X,40A1)
      30  FORMAT(16I5)
      40  CONTINUE
C*****
C*****
C*****
      CREATE NEW FILE FOR QUANTIZER OUTPUT. WRITE
      THE HEADER.
C*****
C*****
      OPEN(UNIT=2,TYPE='NEW',NAME='QUANT.DAT',CARRIAGECONTROL='LIST')
      50  FORMAT(40A1)
      NSPSAM = NSAMP/2
      WRITE(2,20)INSENT,IRATE,NSPSAM,IUPPR,ILOWR,NTERMS,HD
C*****
C*****
      DEFINE AND INITIALIZE VARIOUS PARAMETERS.
C*****

```

```

C*****
K1=773
K2=119
DO 60 JJ=1,100
    CALL RANDU(K1,K2,XX)
60 NUMP = 0
    NADD = 0
    ISEC=0
    IFSC=0
C*****
C*****INITIALIZE THE FVL SHIFTING PARAMETERS.
C*****
READ(8,*)INF,ICF,IAF,IQ1F,IQ2F,IQ3F
INF=2**INF
IQ1F=2**IQ1F
IQ2F=2**IQ2F
IQ3F=2**IQ3F
ISF=2**IS
ITF=ISF-1
ICF=2**ICF
IAF=2**IAF
READ(8,*)NBL
READ(8,*)CLPP
IA4=CLPP*FLOAT(ICF)
IB4=ABS(IA4)
IF(IB4.GT.ITF)TYPE *, ' OVERFLOW INITIAL ICLPP'
ICLPP=FLOAT(IA4)
READ(8,*)KBLK1,ITHIN,ITHMAX
C*****
C*****INITIALIZATION SUBROUTINE.
C*****
CALL INSTRY
C*****
C*****FILE MANAGEMENT FOR RUNS.
C*****
IF(ITCS.NE.1)GO TO 100
OPEN(UNIT=10,TYPE='NEW',NAME='TCLPSP.DAT',CARRIAGECONTROL='LIST')
WRITE(10,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
100 IF(ITSE.NE.1)GO TO 200
OPEN(UNIT=7,TYPE='NEW',NAME='TSENGY.DAT',CARRIAGECONTROL='LIST')
WRITE(7,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
200 IF(ITPF.NE.1)AND(IPATH.NE.1)AND(IPATH.NE.3)GO TO 300
OPEN(UNIT=4,TYPE='NEW',NAME='TPITCH.DAT',CARRIAGECONTROL='LIST')
WRITE(4,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
300 IF(ITFF.NE.1)GO TO 400
OPEN(UNIT=11,TYPE='NEW',NAME='TFILTER.DAT',CARRIAGECONTROL='LIST')
WRITE(11,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
400 IF(IPATH.NE.3)AND(IPATH.NE.4)GO TO 500
OPEN(UNIT=12,TYPE='NEW',NAME='TZHAT.DAT',CARRIAGECONTROL='LIST')
WRITE(12,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
500 CONTINUE
C*****

```

```

C*****
C*****
C*****BEGIN MAIN LOOP. SPEECH BUFFER CREATED AND CLIPPED
C*****SPEECH GENERATED BY SUBROUTINE.
C*****
      ISENGV=0
      BUFSZ=2*ITMAX
      BUFSZ=(BUFSZ/16)*16+16
      READ(1,30)(ISBUF(I),I=1,BUFSZ)
      DO 44 I=1,BUFSZ
        ISBUF(I)=ISBUF(I)*INF
      44 CONTINUE
      STRT=1
      22 CONTINUE
      IF(NOFILT.EQ.1)GO TO 490
      DO 505 I=STRT,BUFSZ
        IFSP(I)=ISBUF(I)
      505 CONTINUE
      GO TO 485
C*****
C*****LOW PASS FILTER SUBROUTINE.
C*****
      490 CALL IFILT(ISBUF,STRT,BUFSZ,IFSP)
C*****
C*****SUBROUTINE TO CLIP SPEECH.
C*****
      485 CALL CLP(IFSP,STRT,BUFSZ,CLPSP,ISENGV)
C*****
C*****PITCH DETERMINED BY AUTOCORRELATION
C*****
      21 IBIGG=-1000
      12SUM=-1000
      DO 12 IT=ITMIN,ITMAX+1
        ISUM=0
        DO 13 J=1,KBLK1
          IF((CLPSP(J+IT).LT.0).AND.(CLPSP(J).LT.0)).OR.
            ((CLPSP(J+IT).GT.0).AND.(CLPSP(J).GT.0)))
            ISUM=ISUM+1
          IF((CLPSP(J+IT).LT.0).AND.(CLPSP(J).GT.0)).OR.
            ((CLPSP(J+IT).GT.0).AND.(CLPSP(J).LT.0)))
            ISUM=ISUM-1
        13 CONTINUE
        IF(12SUM.GT.IBIGG.AND.12SUM.GT.ISUM)NP=IT-1
        IF(12SUM.GT.IBIGG.AND.12SUM.GT.ISUM)IBIGG=12SUM
        12SUM=ISUM
      12 CONTINUE
      NUMP=NUMP+1
C*****
C*****RANDOM PITCH GENERATOR.
C*****

```

```

NINC=MODN(NUMP)
IF(NP.GT.ITMIN) GO TO 29
  CALL RANDU(K1,K2,XX)
  NP=XX*(ITMAX-ITMIN)/2+ITMIN
  WRITE(5,462)NUMP,NP
29 IPICH(NINC)=NP
462 FORMAT(/5X,'NUMP=',I4,5X,'NP IS=',I4)
  IF(ITPF.NE.1.AND.IPATH.NE.1.AND.IPATH.NE.3)GO TO 558
  IF(NINC.EQ.16)WRITE(4,38)(IPICH(J),J=1,16)
C*****
C*****WINDOW FORMED BY SUBROUTINE FOR SPEECH COMPRESSION
C*****FROM PITCH GENERATED.
C*****
558 CALL WINDOW(H,NP)
C*****
C*****
C*****SPEECH IS COMPRESSED AND ARC IS EMPLOYED.
C*****
DO 14 I=1,NP
  NUM4=NP+I
  IARG=NADD+I
  N1=MODN(IARG)
  IA4=ISBUF(NUM4)
  IB4=ISBUF(I)
  IA4=IB4-IA4
  IB4=ABS(IA4)
  IF(IB4.GT.ITF)TYPE *, ' OVERFLOW COMP ADD '
  IA4=FLOAT(ISBUF(NUM4))+H(I)*FLOAT(IA4)
  IB4=ABS(IA4)
  IF(IB4.GT.ITF)TYPE *, ' OVERFLOW COMP ADD 2 '
  IVY=IA4
  IVY(N1)=IVY/INF
  IF((IPATH.EQ.3.OR.IPATH.EQ.4).AND.N1.EQ.16)WRITE(12,38)
    (IVY(J),J=1,16)
  CALL ARC(IVY,Q,IVHAT)
  JOQ=Q
  STAT(JOQ)=STAT(JOQ)+1
  SQ(N1)=Q
  IF(N1.EQ.16)WRITE(2,33)SQ
14 CONTINUE
C*****
C*****UPDATE FOR NEXT BLOCK OF SPEECH. BUFFERS ARE
C*****SHIFTED OVER 2*NP PLACES AND THEN FILLED AGAIN.
C*****IF THE END OF THE FILE IS REACHED, ZEROS ARE AP-
C*****PENDED TO FINISH PROCESSING THE FILE.
C*****
NADD=NADD+NP
NP2=2*NP
INBUFF=BUF5Z-NP2

```

```

DO 15 I=1,INBUFF
  ISBUF(I)=ISBUF(NP2+I)
  CLPSP(I)=CLPSP(NP2+I)
  IFSP(I)=IFSP(NP2+I)
15 CONTINUE
  BUF SZ=2*ITMAX
  MORE=BUF SZ-INBUFF
  IF(MOD(MORE,16).EQ.0)GO TO 25
  MORE=(MORE/16)*16+16
25 BUF SZ=INBUFF+MORE
  NLEFT=NSAMP-2*NADD
  IF(MORE.GT.NLEFT)GO TO 23
  READ(1,30,END=26)((ISBUF(INBUFF+I),I=1,MORE)
  DO 45 I=1,MORE
    ISBUF(INBUFF+I)=ISBUF(INBUFF+I)*INF
45 CONTINUE
26 STOP-INBUFF+1
  GO TO 22
23 CHK=0.
  DO 17 I=1,INBUFF
    CHK=CHK+FLOAT(ABS(ISBUF(I)))
17 CONTINUE
  IF(CHK.EQ.0.)GO TO 24
  NZEROS=MORE-NLEFT
  READ(1,30,END=27)((ISBUF(INBUFF+I),I=1,NLEFT)
  DO 46 I=1,NLEFT
    ISBUF(INBUFF+I)=ISBUF(INBUFF+I)*INF
46 CONTINUE
27 DO 16 I=1,NZEROS
  ISBUF(INBUFF+NLEFT+I)=0
16 CONTINUE
  STRT=INBUFF+1
  GO TO 22
24 NONES=16-N1
  DO 18 I=1,NONES
    SQ(N1+I)=1
18 CONTINUE
  WRITE(2,33)SQ
  IF(ITPF.EQ.1.OR.IPATH.EQ.1.OR.IPATH.EQ.3)WRITE(4,30)
  1 (IPICH(J),J=1,NINC)
  IF(IPATH.EQ.3)WRITE(12,30)(IVV(J),J=1,N1)
C*****STATISTICS GENERATED FOR OUTPUT FILE BY SUBROUTINE.
C*****
C*****CALL INEND*****
C*****
C*****CLOSE FILES.
C*****
33 FORMAT(16I2)
  WRITE(6,*)JHSZ,ILSZ

```

CLOSE(UNIT=7)  
CLOSE(UNIT=1)  
CLOSE(UNIT=2)  
CLOSE(UNIT=8)  
CLOSE(UNIT=4)  
CLOSE(UNIT=11)  
STOP  
END





```

DO 20 K=1,NO
  TNEV=FLOAT(IT(K))*FLOAT(ISIZE)*SCALE*2.
  20 IF(IXX.GE.TNEV) I=2*K+.5+F
     J=(I+2)/2
C*****
C*****THE VALUE OF THE QUANTIZED ERROR IS DETERMINED.
C*****
  IV=2.*F*FLOAT(IOUT(J))*FLOAT(ISIZE)/FLOAT(ISF)
  SCALE=FLOAT(ISF)*SCALE
  IV=IV*SCALE*2.
C*****
C*****THE STANDARD DEVIATION ESTIMATE IS UPDATED.
C*****
  ISIZE=FLOAT(IEPN(J))*FLOAT(ISIZE)/FLOAT(ISF)
  SCALE=FLOAT(ISF)/FLOAT(IQ2F)
  ISIZE=FLOAT(ISIZE)*SCALE
  IRMS=IRMS/2
  IF(IQ3F.GT.INF) ISIZE=FLOAT(ISIZE)*(FLOAT(INF)/FLOAT(IQ3F))
  IF(INF.GT.IQ3F) IRMS=FLOAT(IRMS)*(FLOAT(IQ3F)/FLOAT(INF))
  ISIZE=IMAX#(ISIZE,IRMS)
  IF(IQ3F.GT.INF) ISIZE=FLOAT(ISIZE)*(FLOAT(IQ3F)/FLOAT(INF))
  IF(INF.GT.IQ3F) ISIZE=FLOAT(ISIZE)*(FLOAT(INF)/FLOAT(IQ3F))
  RETURN
  END

```



```

ILSZ=10000
VARE6=9
READ(8,*)ALP,ALAD,ABIAS,G,N,RMSMIN,SMIN IARC PARAMETERS
IRMSMN=RMSMIN*FLOAT(INF)
2 FORMAT(/6X,'ALP=',F5.2,2X,'ALAD=',F5.2,2X,'G='
1 F5.3,2X,'N=',12,2X,'RMSMIN=',F5.1,2X,'SMIN=',F5.2,2X,'ABIAS='
2 F5.3/)
READ(8,*)KQ
NQ=KQ/2
NQ0Q=NQ+1
C*****
C*****
C*****
C*****
C*****QUANTIZER VALUES ARE SET AND APPROPRIATELY
C*****SHIFTED.
C*****
READ(8,*)(T(I),I=1,NQ0Q) ITHRESHOLD VALUES
READ(8,*)(OUT(I),I=1,NQ0Q) ISCALING FACTORS
READ(8,*)(EXP(I),I=1,NQ0Q) IEXPANSION FACTORS
DO 455 I=1,NQ0Q
IOUT(I)=OUT(I)*FLOAT(IQIF)*SMIN
IEXPN(I)=EXPN(I)*FLOAT(IQ2F)
IT(I)=T(I)*FLOAT(IQIF)*SMIN
455 CONTINUE
READ(8,*)ALPHA ISIGNAL ENERGY PARAMETER
C*****FILE MANAGEMENT FLAGS.
READ(8,*)ITCS,ITPF,ITSE,ITSF,IRCS,IRPF,IRSE,ICSF,IOSF,ITFF,IRFF
READ(8,*)NOFLT IFILTER FLAG
READ(8,*)(HF(I),I=1,5) IFILTER COEFFICIENTS
READ(8,*)IPATH IDETERMINES MAJOR ROUTE OF PROGRAM
READ(8,*)(B(I),I=1,N) IINITIAL PREDICTOR COEFFICIENTS
READ(8,*)GAIN
C*****
C*****
C*****
C*****OPEN STATISTICS FILE AND WRITE PARAMETERS.
C*****
IF(ITSF.NE.1)GO TO 201
OPEN(UNIT=9,TYPE='NEW',NAME='TSTATS.DAT')
WRITE(VARE6,2)ALP,ALAD,G,N,RMSMIN,SMIN,ABIAS
WRITE(VARE6,16)KQ,NQ0Q
165 FORMAT(/' THE FOLLOWING IS A LIST OF SCALING FACTORS, '
1 ' EXPANSION FACTORS, AND THRESHOLDS FOR A ',12,' LEVEL' /
2 ' QUANTIZER(',12,' ON A SIDE)')//
WRITE(VARE6,18)
DO 195 I=1,NQ0Q
WRITE(VARE6,20)I,OUT(I),T(I),EXPN(I)
195 CONTINUE
185 FORMAT(/3X,'LEVEL NUMBER',4X,'SCALING FACTORS',4X,'THRESHOLDS',
1 4X,'EXPANSION FACTORS')/
205 FORMAT(6X,12,17X,F7.4,9X,F7.4,10X,F7.4)
201 ISIZE=1000*IQ3F IINITIAL STANDARD DEVIATION ESTIMATE

```

```

DO 118 I=1,12
  IVHAT(I)=S
  IV(I)=S
  B(I)=B(I)*FLOAT(IAF)
  IA(I)=B(I)
118 CONTINUE
  ABLAS=ABLAS*FLOAT(IAF)
  DO 119 I=1,9
    IX(I)=S
119 CONTINUE
  IRMS=IRMSH
  C*****
  C*****ENERGY VALUES FOR STATISTICS ARE INITIALLY SET TO
  C*****ZERO.
  C*****
  EV=S.
  EP=S.
  ENGY1=S.
  ENGY2=S.
  J1=S
  ENGY3=S.
  ENGY4=S.
  ENGY5=S.
  ENGY6=S.
  ENGY7=S.
  ENGY8=S.
  NDIV=S
  SPER1=S.
  SPER2=S.
  DO 621 I=1,KO
    ISTAT1(I)=S
621 RETURN
  END

```



```

IPRE1=FLOAT(IPRE1)+IA(1)*FLOAT(IV(1))/FLOAT(IAF)
IPRE=FLOAT(IPRE)+IA(1)*FLOAT(IVHAT(1))/FLOAT(IAF)

125 CONTINUE
IRMS=ALP*FLOAT(IRMS-IRMSN)+FLOAT(IRMSN)
IRMS=FLOAT(IRMS)+(1.-ALP)*ABS(FLOAT(IVHAT(1)))
C*****
C*****ERROR SAMPLE GENERATED.
C*****
IERROR=IV-IPRE
IERR1=IV-IPRE1
C*****
C*****QUANTIZER SUBROUTINE IS ENGAGED.
C*****
CALL QUANT(IERROR,EQ,IOUT)
C*****
C*****UPDATE PREDICTOR VALUES.
C*****
ISTAT1(IOUT)=ISTAT1(IOUT)+1
Q=IOUT
DO 125 I=1,N
J=N+2-I
IV(J)=IV(J-1)
IVHAT(J)=IVHAT(J-1)
125 IVHAT(1)=IPRE+EQ
IV(1)=IV
IVHAT=IVHAT(1)
IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,*)IOCNT
IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,*)IV,IPRE,EQ,IVHAT
IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,244)
244 FORMAT(' ')
C*****
C*****
C*****ADAPTATION OF PREDICTOR COEFFICIENTS IS ACCOMPLISHED
C*****BY A FIXED PERCENTAGE COMPUTATION.
C*****
IF(G.EQ.5.5)GO TO 134
DO 135 I=1,N
IA(1)=IA(1)-B(I)
135 CONTINUE
DO 135 I=1,N
SVL=ISIGN(1,EQ)*ISIGN(1,IVHAT(I+1))
IF((EQ.EQ.5.OR.IVHAT(I+1).EQ.5).AND.IA(1).GE.5)SVL=-1.5
IF((EQ.EQ.5.OR.IVHAT(I+1).EQ.5).AND.IA(1).LT.5)SVL=+1.5
AINT=SVL*AMAX1(FLOAT(IABS(IA(1)))*G,ABIAS)
IA(1)=ALAD*(IA(1)+AINT)
135 CONTINUE
DO 155 I=1,N
IA(1)=IA(1)+B(I)
155 CONTINUE
134 IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,*)IRMS
IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,*)(IA(KKK),KKK=1,4)

```

```

IF(IOCNT.GT.165.AND.IOCNT.LE.228)WRITE(6,244)
IF(IOCNT.GT.165.AND.IOCNT.LE.228)WRITE(6,244)
C*****
C*****ENERGY STATISTICS ARE GENERATED FOR DETERMINATION
C*****OF SNR VALUES REPORTED IN STATISTICS FILE.
C*****
IRN=MOD(IARG,NBL)
ENGY1=ENGY1+FLOAT(IY)**2.
ENGY2=ENGY2+(FLOAT(IERROR)-FLOAT(EQ))**2.
ENGY3=ENGY3+FLOAT(IERROR)**2
ENGY4=ENGY4+FLOAT(IERRR1)**2
ENGY5=ENGY5+FLOAT(IY)*FLOAT(IYHAT)
ENGY6=ENGY6+FLOAT(IYHAT)**2
ENGY7=ENGY7+FLOAT(IYHAT)
ENGY8=ENGY8+FLOAT(IY)
NDIV=NDIV+1
IF(IRN.NE.8)GOTO 133
JI=JI+1
EV=EV+ENGY1
EP=EP+ENGY2
IF(ENGY3.NE.8.AND.ENGY1.NE.8.)SPERB1(JI)=18.*ALOG18(ENGY1/ENGY3)
IF(ENGY4.NE.8.AND.ENGY1.NE.8.)SPERB2(JI)=18.*ALOG18(ENGY1/ENGY4)
IF(ENGY2.NE.8.AND.ENGY1.NE.8.)SNRB(JI)=18.*ALOG18(ENGY1/ENGY2)
IF(ENGY2.NE.8.AND.ENGY3.NE.8.)SNRQB(JI)=18.*ALOG18(ENGY3/ENGY2)
SPER1=SPER1+ENGY3
SPER2=SPER2+ENGY4
ENGY1=8.
ENGY2=8.
ENGY3=8.
ENGY4=8.
CONTINUE
RETURN
END

```

133

[illegible]

```

C*****
IF(ITSF.NE.1)RETURN
SNR=10.*ALOG10(EV/EP)
SPER=10.*ALOG10(EV/SPER1)
SPER1=10.*ALOG10(EV/SPER2)
SNRQ=10.*ALOG10(SPER1/EP)
ISUM=0
DO 300 I=1,KQ
  ISUM=ISUM+ISTAT1(I)
300 ARG1=ISUM
  SUM=ALOG(ARG1)
  DO 500 I=1,KQ
    ARG=ISTAT1(I)+0.001
    SUM=SUM-(ARG*ALOG(ARG))/ARG1
    BITS=SUM/ALOG(2.)
    WRITE(VARE6,402)SNR,BITS,(ISTAT1(I),I=1,KQ)
402 FORMAT(6X,'SNR INLOOP=',F7.3X,'H=',F4.2X,'OP',
1 1015.3(/22X,1015)///)
    IF(VARE6.EQ.6)WRITE(5,402)SNR,BITS,(ISTAT1(I),I=1,KQ)
    WRITE(VARE6,404)
    FORMAT(///6X,'SAMPLE NUMBER ',3X,' SNR ',7X,' SPER ',5X,
404 1 'SPER1 ',4X,' SNRQ '///)
    NB=ISUM/NBL
    DO 400 I=1,NB
      IS=(I-1)*NBL+1
      IE=IS+NBL-1
      WRITE(VARE6,412)IS,IE,SNRB(I),SPERB1(I),SPERB2(I),SNROB(I)
400 CONTINUE
      FORMAT(6X,15,'-',15,6X,F7.2,4X,F7.2,4X,F7.2,4X,F7.2)
      WRITE(VARE6,416)SPER,SPER1,SNRO
416 FORMAT(//6X,'PREDICTOR PERFORMANCE =',F8.2/6X,
1 'PREDICTOR IDEAL PERFORMANCE=',F8.2/6X,' SIGNAL TO NOISE
2 RATIO=',F8.2)
    NQUA=0
    DO 410 I=1,32
      NQUA=NQUA+STAT(I)
410 DO 420 I=1,32
      PROB(I)=FLOAT(STAT(I))/FLOAT(NQUA)
420 CONTINUE
      WRITE(VARE6,422)
422 FORMAT(//.6X,'LEVEL NUMBER',6X,'PROBABILITY',6X,' FREQUENCY',/)
424 FORMAT(9X,12,12X,F7.4,10X,15)
      DO 426 I=1,32
        WRITE(VARE6,424)I,PROB(I),STAT(I)
426 CONTINUE
      RETURN
      END

```



AD-A135 232

REAL-TIME IMPLEMENTATION OF A SPEECH DIGITIZATION  
ALGORITHM COMBINING TIM. (U) NOTRE DAME UNIV IN DEPT OF  
ELECTRICAL ENGINEERING J L MELSA ET AL. JUN 83

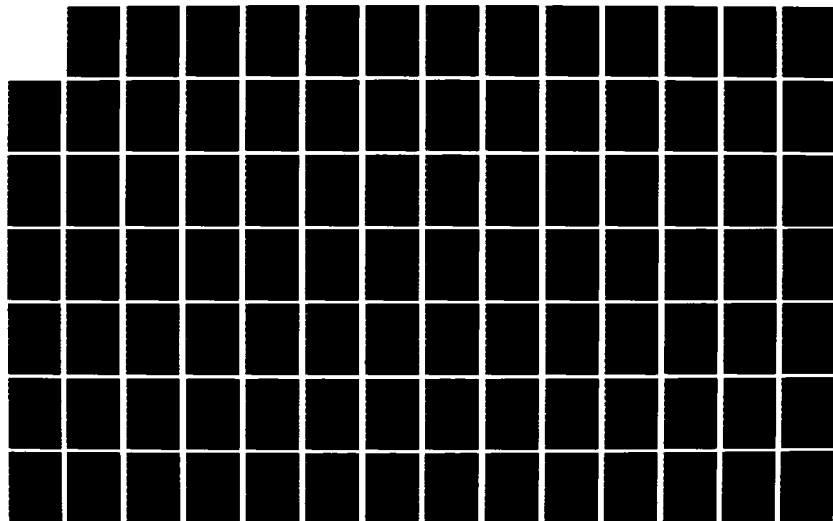
2/2

UNCLASSIFIED

DCA100-82-C-0026

F/G 9/4

NL





```

28      IA4=FLOAT(ICLPP)*FLOAT(ISENGY)/ICF
      IB4=ABS(IA4)
      IF (IB4.GT.1TF) TYPE = ' OVERFLOW CLPP '
      CLPLVL=FLOAT(IA4)
      CLM=-CLPLVL
      CLPSP(1)=0
      IF (1FLTR.GT.CLPLVL)CLPSP(1)=+1
      IF (1FLTR.LT.CLW)CLPSP(1)=-1
      CLIPFI(MSEC)=CLPSP(1)
      IF (ITCS.NE.1)GO TO 21
      IF (MSEC.EQ.16)WRITE(18,18)(CLIPFI(JJ),JJ=1,16)

21 CONTINUE
18 FORMAT(16I5)
      RETURN
      END

```





2000

**TITLE: COK.FTN**

DESCRIPTION: THIS PROGRAM INDUCES ERRORS INTO THE SEQUENCE OF QUANTIZATION LEVELS. IT IS RES-TRICTED TO INDUCING ONLY ONE ERROR PER CODEWORD (FOUR BIT STRING) AND OPERATES AT AN INTERACTIVE BER. IT IS TO BE USED AT A 16KHZ TRANSMISSION RATE AND AN 8KHZ SAMPLED SPEECH RATE TDHS-ARC SYSTEM SINCE IT WILL INDUCE ERRORS ONLY IN THE FIRST 15 LEVELS. THE CODEWORD ASSIGNMENTS WERE CHOSEN BY ARVIND ARORA AND ARE BASED ON A PSEUDO-MSE FOR A ONE BIT ERROR PER CODEWORD SCHEME.

**DIMENSION IN(16), IOUT(16)**

\*\*\*\*\*FORMATS USED.

15 FORMAT(16I2)

```
12# FORMAT(' * ERRORS= ',F5.0,3X,F5.0,4X,F5.0,3X,F5.0)
```

30 FORMAT(6I5,10X,45A1)

```
4# FORMAT('X ERRORS= ',F5.4,3X,F5.4,4X,F5.4,3X,F5.4)
```

50 FORMAT(' TOTAL ERROR = ',F5.4,' COMPARED TO DESIGN = ',F5.4)

\*\*\*\*\*FILE MANAGEMENT.

```
OPEN(UNIT=2,TYPE='NEW',NAME='CSTATS.DAT')
```

```
OPEN(UNIT=3,TYPE='OLD',NAME='QUANT.DAT',
```

```
OPEN(UNIT=3, FILE=OLD, STATUS='OLD', NAME='QUANT.DAT',  
CARRIAGECONTROL='LIST')  
OPEN(UNIT=4, TYPE='NEW', NAME='DECO.DAT', CARRIAGECONTROL='LIST')
```

READ(3,3) NSENT, IRATE, NSAMP, IUPPR, ILOWR, NTERMS, HD







မမမမမမ

**SECRET**





```

IAF=200IAF
IQ1F=200IQ1F
IQ2F=200IQ2F
IQ3F=200IQ3F
ISF=200ISF
ITF=ISF-1
READ(4,*)NBL
READ(4,*)CLPP
ICLPP=CLPP*FLOAT(ICF)
READ(4,*)KBLK1,ITWIN,ITMAX
C*****INITIALIZATION SUBROUTINE.
C*****
C*****CALL INSTRT
C*****
C*****
C*****
C*****OPEN FILES AND WRITE HEADERS.
C*****
      OPEN(UNIT=1,TYPE='OLD',NAME='DECO.DAT')
      READ(1,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTURNS,HD
      NSPSAM=NSAMP*NSAMP
      IF(ICSF.NE.1)GO TO 405
      OPEN(UNIT=3,TYPE='NEW',NAME='ZHAT.DAT',CARRIAGECONTROL='LIST')
      WRITE(3,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTURNS,HD
      405 IF(ICSF.NE.1)GO TO 415
      OPEN(UNIT=7,TYPE='NEW',NAME='SHAT.DAT',CARRIAGECONTROL='LIST')
      WRITE(7,20)NSENT,IRATE,NSPSAM,IUPPR,ILOWR,NTURNS,HD
      415 IF(IRSE.NE.1)GO TO 425
      OPEN(UNIT=9,TYPE='NEW',NAME='RSENGY.DAT',CARRIAGECONTROL='LIST')
      WRITE(9,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTURNS,HD
      425 IF(IRPF.NE.1)GO TO 435
      OPEN(UNIT=2,TYPE='NEW',NAME='RPITCH.DAT',CARRIAGECONTROL='LIST')
      WRITE(2,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTURNS,HD
      435 IF(IRCS.NE.1)GO TO 445
      OPEN(UNIT=10,TYPE='NEW',NAME='RCLPSP.DAT',CARRIAGECONTROL='LIST')
      WRITE(10,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTURNS,HD
      445 IF(IRFF.NE.1)GO TO 445
      OPEN(UNIT=11,TYPE='NEW',NAME='REFILTER.DAT',CARRIAGECONTROL='LIST')
      WRITE(11,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTURNS,HD
      445 IF(IPATH.EQ.1.OR.IPATHEQ.2)GO TO 435
      OPEN(UNIT=12,TYPE='OLD',NAME='TZHAT.DAT')
      READ(12,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTURNS,HD
      435 IF(IPATH.EQ.2.OR.IPATHEQ.4)GO TO 425
      OPEN(UNIT=13,TYPE='OLD',NAME='TPITCH.DAT')
      READ(13,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTURNS,HD
      425 CONTINUE
C*****
C*****
C*****INITIALIZE REMAINING PARAMETERS.
C*****

```



```

500 CONTINUE
  IF(IPATH.EQ.1.OR.IPATH.EQ.3)GO TO 350
C*****COMPRESSED SPEECH IS FILTERED BY SUBROUTINE.
C*****
  IF(NOFLT.EQ.1)GO TO 490
  DO 505 I=1,ISTR1,IFNSH1
    IFS(I)=QCOMP(I)
505 CONTINUE
    GO TO 485
  490 CALL IFLT(QCOMP,ISTR1,IFNSH1,IFS)
C*****CLIP THE COMPRESSED SPEECH BY SUBROUTINE.
C*****
  485 CALL CLPPP(IFS,ISTR1,IFNSH1,CLPSP,ISENGY)
C*****
C*****PITCH DETERMINED BY AUTOCORRELATION.
C*****
  510 IBIGG=-1000
  12SUM=-1000
  DO 520 IT=ITMIN,ITMAX+1
    ISUM=0
    IBGN=ITMAX
    ISTD=3*KBLK1
    DO 525 J=IBGN,ISTD
      IF(((CLPSP(J-IT).LT.#).AND.(CLPSP(J).LT.#)).OR.
        ((CLPSP(J-IT).GT.#).AND.(CLPSP(J).GT.#)))
        ISUM=ISUM+1
      IF(((CLPSP(J-IT).LT.#).AND.(CLPSP(J).GT.#)).OR.
        ((CLPSP(J-IT).GT.#).AND.(CLPSP(J).LT.#)))
        ISUM=ISUM-1
    525 CONTINUE
    IF(12SUM.GT.IBIGG.AND.12SUM.GT.ISUM)NP=IT-1
    IF(12SUM.GT.IBIGG.AND.12SUM.GT.ISUM)IBIGG=12SUM
    12SUM=ISUM
520 CONTINUE
  350 JJ=JJ+1
C*****RANDOM PITCH GENERATOR.
C*****
  JINC=MODN(JJ)
  IF(IPATH.EQ.2.OR.IPATH.EQ.4)GO TO 355
  IF(JINC.EQ.1)READ(13,50,END=365)(IPICH(J),J=1,16)
  NP=IPICH(JINC)
  GO TO 515
360 NP=ITMIN
355 IF(NP.GT.ITMIN)GO TO 515
  CALL RANDU(K1,K2,XX)
  NP=XX*(ITMAX-ITMIN)/2+ITMIN
  WRITE(5,200)JJ,NP

```

```

515 IPICH(JINC)=NP
288 FORMAT(6X,'NUMP=',I4.6X,'NP IS=',I4.12X,'RANDOM PITCH!!!!!!')
IF(IOPF.NE.1)GO TO 535
IF(JINC.EQ.16)WRITE(2,55)(IPICH(K),K=1,16)
535 NP2=NP+NP
C*****
C*****WINDOW FORMED FOR SPEECH EXPANSION
C*****FROM PITCH GENERATED.
C*****
DO 545 I=1,NP2
545 H(I)=1.-FLOAT(I-1)/FLOAT(NP2-1)
545 CONTINUE
1
C*****
C*****SPEECH IS EXPANDED.
C*****
555 P1=ITMAX-NP
NP3=NP2+NP
DO 555 I=1,NP3
IVHAT(I)=QCOMP(P1+I)
555 CONTINUE
DO 565 I=1,NP2
IARG=NADD+I
N1=MODN(IARG)
IA4=IVHAT(I+NP)
IB4=IVHAT(I)
IA4=IA4-IB4
IB4=ABS(IA4)
IF(IA4.GT.IIF)TYPE=,' OVERFLOW EXPN ADD'
IA4=FLOAT(IVHAT(I))+H(I)*FLOAT(IA4)
IB4=ABS(IA4)
IF(IA4.GT.IIF)TYPE=,' OVERFLOW EXPN ADD 2'
SQ(N1)=FLOAT(IA4)/FLOAT(INF)
IF(IOSF.NE.1)GO TO 565
IF(N1.EQ.16)WRITE(7,55)(SQ(J),J=1,16)
565 CONTINUE
C*****
C*****
C*****SHIFT ALL VECTORS APPROPRIATELY BASED ON PITCH
C*****VALUE USED TO EXPAND SPEECH. SET UP VECTORS
C*****AND PARAMETERS FOR NEXT RUN THOROUGH MAIN LOOP.
C*****
NADD=NADD+NP+NP
INC3=ITMAX-NP+NNEED+NLEFT
ISTR1=INC3+1
NLEFT=NNEED-NP+NLEFT
DO 565 I=1,INC3
QCOMP(I)=QCOMP(I+NP)
CLPSP(I)=CLPSP(I+NP)

```

```

      IFS(1)=IFS(1+NP)
565 CONTINUE
      NNEED=MODM(NP)
      NTOT=INC3+NNEED
      IF(NTOT.GT.NSTOP)NNEED=NNEED-16
      IF(N1=INC3+NNEED
      IF(NADD.GE.NSPSAM)GO TO 600
      IF(IPATH.LE.2)GO TO 595
      READ(12,55,END=585)(Q(I),I=1,NNEED)
      DO 575 I=1,NNEED
        Q(I)=Q(I)*INF
      CONTINUE
575 IF(IPATH.LE.2)READ(1,45,END=585)(Q(I),I=1,NNEED)
      GO TO 595
585 DO 585 J=1,NNEED
      Q(I)=Q
585 CONTINUE
595 GO TO 495
C*****
C*****AT END OF FILE, FINISH WRITING VALUES TO APPROPRIATE
C*****FILES.
C*****
600 IF(10SF.NE.1)GO TO 610
      WRITE(7,50)(SQ(J),J=1,M1)
610 IF(1RSE.NE.1)GO TO 620
      WRITE(9,50)(ISENG(J),J=1,INC1)
620 IF(1CSF.NE.1)GO TO 630
      WRITE(3,50)(IZHAT(J),J=1,INC1)
630 IF(1RPF.NE.1)GO TO 640
      WRITE(2,50)(IPICH(J),J=1,JINC)
640 IF(1RCS.NE.1)GO TO 650
      WRITE(10,50)(CLPSP(J),J=1,MSEC)
650 IF(1RFF.NE.1)GO TO 670
      WRITE(11,50)(IFS(J),J=1,IFINC)
670 CONTINUE
      WRITE(5,*)IFSC,YL,YH
C*****
C*****CLOSE FILES.
C*****
      CLOSE(UNIT=1)
      CLOSE(UNIT=3)
      CLOSE(UNIT=4)
      CLOSE(UNIT=7)
      CLOSE(UNIT=2)
      CLOSE(UNIT=11)
      CLOSE(UNIT=10)
      STOP
      END

```



```

C*****
  ISIZE=FLOAT(IEXP(J))*FLOAT(ISIZE)/FLOAT(ISF)
  SCALE=FLOAT(ISF)/FLOAT(IQ2F)
  ISIZE=FLOAT(ISIZE)*SCALE
  IRMS=IRMS/2.
  IF(IQ3F.GT.INF) ISIZE=FLOAT(ISIZE)*(FLOAT(INF)/FLOAT(IQ3F))
  IF(INF.GT.IQ3F) IRMS=FLOAT(IRMS)*(FLOAT(IQ3F)/FLOAT(INF))
  ISIZE=IMAXB(ISIZE,IRMS)
  IF(IQ3F.GT.INF) ISIZE=FLOAT(ISIZE)*(FLOAT(IQ3F)/FLOAT(INF))
  IF(INF.GT.IQ3F) ISIZE=FLOAT(ISIZE)*(FLOAT(INF)/FLOAT(IQ3F))
  RETURN
  END

```



```

READ(4,*)KQ
NQ=KQ/2
NOOO=NO+1
C*****
C*****
C*****QUANTIZER VALUES ARE SET AND APPROPRIATELY
C*****SHIFTED.
C*****
READ(4,*)(T(I),I=1,NOOO)
READ(4,*)(OUT(I),I=1,NOOO)
READ(4,*)(EXP(I),I=1,NOOO) IEXPANSION FACTORS
DO 400 I=1,NOOO
IOUT(I)=OUT(I)*FLOAT(IQ1F)*SHIN
IEXPN(I)=EXPN(I)*FLOAT(IQ2F)
400 CONTINUE
READ(4,*)ALPHA ISIGNAL ENERGY PARAMETER
C*****FILE MANAGEMENT FLAGS.
READ(4,*)ITCS,ITPF,ITSE,ITSF,IRCS,IRPF,IRSE,ICSF,IOSF,ITFF,IRFF
READ(4,*)NOFLT IFILTER FLAG
READ(4,*)(NF(I),I=1,5) IFILTER COEFFICIENTS
READ(4,*)IPATH IPATH DETERMINES MAJOR ROUTE OF PROGRAM
READ(4,*)(B(I),I=1,N) INITIAL PREDICTOR COEFFICIENTS
READ(4,*)GAIN
ISIZE=100*IQ3F INITIAL STANDARD DEVIATION ESTIMATE
DO 110 I=1,12
IVNAT(I)=0
B(I)=B(I)*FLOAT(IAF)
IA(I)=B(I)
110 CONTINUE
ABIAS=ABIAS*FLOAT(IAF)
DO 119 I=1,9
IXXX(I)=0
119 CONTINUE
IRMS=IRMSHN
RETURN
END

```



```

C*****
C*****ERROR SAMPLE GENERATED BY SUBROUTINE.
C*****
C*****
C*****CALL INVQUA(Q,IEQ)
C*****
C*****UPDATE PREDICTOR VALUES.
C*****
      DO 125 I=1,N
      J=M+2-1
125  IVHAT(J)=IVHAT(J-1)
      IVHAT(1)=IPRE+IEQ
      IVHAT1=IVHAT(1)
      IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,*)IOCNT
      IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,*)IPRE,IEQ,IVHAT1
      IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,244)
244  FORMAT(
C*****
C*****
C*****ADAPTATION OF PREDICTOR COEFFICIENTS IS ACCOMPLISHED
C*****BY A FIXED PERCENTAGE COMPUTATION.
C*****
      IF(G.EQ.5.5)GO TO 131
      DO 155 I=1,N
      IA(I)=IA(I)-B(I)
155  CONTINUE
      DO 135 I=1,N
      SVL=ISIGN(1,IEQ)*ISIGN(1,IVHAT(I+1))
      IF((IEQ.EQ.5.OR.IVHAT(I+1).EQ.5).AND.IA(I).GE.5)SVL=-1.
      IF((IEQ.EQ.5.OR.IVHAT(I+1).EQ.5).AND.IA(I).LT.5)SVL=+1.
      AINT=SVL*AMAX1(FLOAT(IABS(IA(I))),5,ABIAS)
      IA(I)=ALAD*(IA(I)+AINT)
135  CONTINUE
      DO 155 I=1,N
      IA(I)=IA(I)+B(I)
155  CONTINUE
      IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,*)ERR,IRMS
      IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,*)(IA(I),I=1,4)
      IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,244)
      IF(IOCNT.GT.165.AND.IOCNT.LE.225)WRITE(6,244)
131  RETURN
      END

```

[illegible]

```

SCALE=FLOAT(15F)/FLOAT(ICF)
IA4=FLOAT(IA4)*SCALE
IB4=ABS(IA4)
IF(1B4.GT.1TF)TYPE *,' OVERFLOW CLPP'
CLPLVL=FLOAT(IA4)
CLM=-CLPLVL
CLPSP(1)=0
IF(1FSP.GT.CLPLVL)CLPSP(1)=+1
IF(1FSP.LT.CLPLVL)CLPSP(1)=-1
CLIPFI(MSEC)=CLPSP(1)
IF(1RCS.NE.1)GO TO 21
IF(MSEC.EQ.16)WRITE(1B,1B)(CLIPFI(JJ),JJ=1,16)

21 CONTINUE
1B FORMAT(16I5)
RETURN
END

```





APPENDIX B

UNRESTRICTED WORD LENGTH SOURCE FILE



```

C
C
      INTEGER VARE,UU,VV,ANS1(6),CHK,ANS2(3),ANS(9),ANS3(8)
      INTEGER*2 FNAME(16)
      REAL MSF,LFI(51),LMX,LOEXP,N,HIEXP,N,HIDEL,T,LODEL,EXP,N(51),M
      REAL OUT(51),KK,LL,T(51),HF(5),H(5,4),AB(12)
C*****
C*****THE FOLLOWING DATA IS THE FILTER COEFFICIENTS FOR
C*****THE FILTERS THAT ARE INTERACTIVELY EMPLOYED BY
C*****THE SIMULATIONS.
C*****
      DATA H/.0341,.0125,.0139,.0149,.0151,.0560,.0720,.1009,
      1      .1219,.1295,.0376,.0214,.0252,.0277,.0285,.1549,.2344,
      2      .3188,.3356/
C*****
C*****PARAM.DAT IS THE OUTPUT FILE; CONTAINS THE PARAMETERS
C*****FOR THE SIMULATIONS.
C*****
      OPEN(UNIT=3,TYPE='NEW',NAME='PARAM.DAT',CARRIAGECONTROL='LIST')
      TYPE *,* ENTER THE ORIGINAL SPEECH FILE NAME:[32CHAR]
      ACCEPT 9,FNAME
      9 FORMAT(16A2)
C*****
C*****INITIALIZATION OF ALL PROGRAM PARAMETERS.
C*****
      NBL=64
      ALPHA=.992
      ALP=.97
      ALAD=.95
      ABIAS=.001
      G=.015
      N=4
      RMSMIN=35.0
      SHMIN=.3
      MSF=.8
      KQ=15
      B=.9
      LOEXP=.5
      HIEXP=5.0
      CLPP=1.5
      KBLK1=110
      ITMIN=30
      ITMAX=110
      ITCS=.0
      ITPF=.0
      ITSE=.0
      ITSF=.0
      IRCS=.0
      IRPF=.0
      IRSE=.0

```

```

ICSF=9
ITFF=9
IRFF=9
IOSF=1
AB(1)=1.793182
AB(2)=-1.486654
AB(3)=9.566823
AB(4)=-9.146576
DO 53 I=5,12
      AB(I)=9.9
53 CONTINUE
GAIN=9.959
C*****
C*****
C*****
C*****THE FOLLOWING BLOCK SETS UP THE LISTED PARAMETERS
C*****
799 DO 795 I=1,9
      ANS(I)=9
795 CONTINUE
TYPE *, , THE FOLLOWING ARE THE CURRENT VALUES'
WRITE(5,100)ALP,ABIAS,ALPHA,G,N,RMSMIN,SMIN,CLPP,KBLK1,ITMIN,
1 ITMAX,ALAD,NBL
100 FORMAT(/6X,'ALP=',F6.3/6X,'ABIAS=',F6.3/6X,'ALPHA=',F6.3/6X,'G='
,F6.3/6X,'N=',F6.3/6X,'RMSMIN=',F5.1/6X,'SMIN=',F5.2/6X,'CLPP='
,F5.2/6X,'KBLK1=',F5.1/6X,'ITMIN=',F5.1/6X,'ITMAX=',F5.1/6X,'ALAD='
,F5.3/6X,'NBL=',F5.3/6X)
2 TYPE *, , WHICH DO YOU WANT TO CHANGE?'
3 TYPE *, , NO (MORE) CHANGES'
TYPE *, , 1-ALP.
TYPE *, , 2-ABIAS.
TYPE *, , 3-ALPHA.
TYPE *, , 4-G.
TYPE *, , 5-N.
TYPE *, , 6-RMSMIN.
TYPE *, , 7-SMIN.
TYPE *, , 8-CLPP.
TYPE *, , 9-KBLK1,ITMIN,ITMAX.
TYPE *, , TYPE--[SELECTION] (SELECTION) (SELECTION)...--UP TO 9 CHOICES.
ACCEPT 789,ANS
789 FORMAT(9(11,1X))
CHK=9
DO 779 I=1,9
      CHK=CHK+ANS(I)
779 CONTINUE
IF(CHK.EQ.9)GO TO 999
ISELEC=1
197 DO 199 I=1,9
      IF(ANS(I).EQ.ISELEC)GO TO (291,292,293,294,295,296,297,298
,299),ISELEC
199 CONTINUE
GO TO 193

```

```

281 TYPE *, ' ENTER NEW VALUE OF "ALP"
ACCEPT *,ALP
GO TO 193
282 TYPE *, ' ENTER NEW VALUE OF "ABIAS"
ACCEPT *,ABIAS
GO TO 193
283 TYPE *, ' ENTER NEW VALUE OF "ALPHA"
ACCEPT *,ALPHA
GO TO 193
284 TYPE *, ' ENTER NEW VALUE OF "G"
ACCEPT *,G
GO TO 193
285 TYPE *, ' ENTER NEW VALUE OF "N"
ACCEPT *,N
GO TO 193
286 TYPE *, ' ENTER NEW VALUE OF "RMSMIN"
ACCEPT *,RMSMIN
GO TO 193
287 TYPE *, ' ENTER NEW VALUE OF "SMIN"
ACCEPT *,SMIN
GO TO 193
288 TYPE *, ' ENTER NEW VALUE OF "CLPP"
ACCEPT *,CLPP
GO TO 193
289 TYPE *, ' ENTER NEW VALUES OF "KBLK1 ITMIN ITMAX"
ACCEPT *,KBLK1,ITMIN,ITMAX
193 ISELEC=ISELEC+1
IF(1SELEC.LE.9)GO TO 197
GO TO 798
998 CONTINUE
C*****THE FOLLOWING BLOCK SETS UP THE PARAMETERS OF THE
C*****ADAPTIVE RESIDUAL CODER. IT GENERATES THE THRESH-
C*****HOLDS, SCALING FACTORS, AND EXPANSION FACTORS USING
C*****THE PARAMETERS SPECIFIED BY THE USER.
C*****
12 DO 17 I=1,6
17 CONTINUE
DO 18 I=1,6
IF(ANS1(I).EQ.1)GO TO 3
18 CONTINUE
GO TO 18
3 TYPE *, 'ENTER THE MAXIMUM SCALING FACTOR(>8)'
ACCEPT *,MSF
IF (MSF.LE.8.)GO TO 3
18 DO 19 I=1,6
IF(ANS1(I).EQ.2) GO TO 29
19 CONTINUE
GO TO 28

```

```

29 TYPE *, 'ENTER THE NUMBER OF QUANT LEVELS (5<NUM<151) AND ODD'
ACCEPT *, KQ
IF (KQ.LT.5.OR.KQ.GT.151)GO TO 15
25 NQ=KQ/2
NQOQ=NQ+1
C*****
C*****DETERMINATION OF THE QUANTIZER PARAMETER VALUES.
C*****FIRST, THE SCALING FACTORS ARE GENERATED.
C*****
OUT(NQOQ)=MSF
LMX=(ALOG15(MSF))
UP=LMX/(FLOAT(NQ))
LFI(1)=5.
DO 45 I=2,NQ
LFI(I)=LFI(I-1)+UP
45 CONTINUE
KK=MSF+1
KK=ALOG15(KK)
LL=LFI(NQ)
KK=KK/LL
BMAX=15.**KK
DO 214 I=1,5
IF(ANS1(I).EQ.3)GO TO 213
214 CONTINUE
GO TO 42
213 WRITE(5,212)BMAX,B
212 FORMAT(' THE MAX BASE ALLOWED IS '.F5.2,' AND THE CURRENT BASE '
1 ' IS '.F5.2)
TYPE *, 'ENTER BASE OF LOG FNC USED FOR SCALING FACTORS(>5)'
ACCEPT *, B
42 IF(B.GT.5.AND.B.LT.BMAX)GO TO 43
TYPE *, 'THE BASE CHOSEN WAS OUTSIDE THE ALLOWED RANGE. TRY AGAIN!'
GO TO 213
43 DO 55 I=1,NQ
M=LFI(I)
OUT(I)=B**M-1.
55 CONTINUE
DO 65 I=1,NQ
T(I)=(OUT(I)+OUT(I+1))/2.
65 CONTINUE
DO 61 I=1,6
IF(ANS1(I).EQ.4)GO TO 62
61 CONTINUE
GO TO 65
C*****
C*****DETERMINATION OF EXPANSION FACTORS.
62 TYPE *, 'ENTER THE LOW EXPANSION FACTOR'
ACCEPT *, LOEXPN
65 DO 66 I=1,6
IF(ANS1(I).EQ.5)GO TO 67
66 CONTINUE
GO TO 75
67 TYPE *, 'ENTER THE HIGH EXPANSION FACTOR'

```

```

ACCEPT *,HIEXP
78 MID=NO/2+1
   EXPN(MID)=1.
   EXPN(MID+1)=1.
   EXPN(MID-1)=1.
   EXPN(1)=LOEXP
   EXPN(NQOQ)=HIEXP
   HIDEIT=(HIEXP-1.)/(FLOAT(MID-1))
   LOELT=(1.-LOEXP)/(FLOAT(MID-2))
   XX=HIDEIT+1.
   VV=LOELT+LOEXP
   UU=MID-2
   VV=MID+2
   DO 115 I=2,UU
      EXPN(I)=VV
      VV=VV+LOELT
115 CONTINUE
   DO 125 I=VV,MID
      EXPN(I)=XX
      XX=XX+HIDEIT
125 CONTINUE
C*****INTERMEDIATE PRINTOUT OF QUANTIZER PARAMETERS.
TYPE *, 'WHERE SHOULD THE PRINTOUT GO?(TERM=5,PRINT=1)'.
ACCEPT *,1
IF (1.EQ.5) VARE=5
IF (1.EQ.1) VARE=6
WRITE(5,155)ALP,ABIAS,ALPHA,G,N,RHSMIN,SMIN,
1   CLPP,KBLK1,ITMIN,ITMAX,ALAD,NBL
WRITE(5,165)KQ,NQOQ,MSF,B,LOEXP,HIEXP
IF(1.EQ.1)WRITE(WARE,155)ALP,ABIAS,ALPHA,G,N,RHSMIN,SMIN,
1   CLPP,KBLK1,ITMIN,ITMAX,ALAD,NBL
IF(1.EQ.1)WRITE(WARE,165)KQ,NQOQ,MSF,B,LOEXP,HIEXP
165 FORMAT(' THE FOLLOWING IS A LIST OF SCALING FACTORS.'/
1   ' EXPANSION FACTORS, AND THRESHOLDS FOR A ',I2,' LEVEL.'/
2   ' QUANTIZER(',I2,' ON A SIDE), WITH A MAXIMUM SCALING FACTOR.'/
3   ' ',F7.4,' A LOG SCALE WITH A BASE=',F7.4,' WAS USED.'/
4   ' THE HIGH AND LOW EXPANSION FACTORS WERE SET AT ',F7.4/
5   ' AND ',F7.4,'.',
WRITE(WARE,185)
DO 195 I=1,NQOQ
   WRITE(WARE,255)I,OUT(I),T(I),EXPN(I)
195 CONTINUE
185 FORMAT('/3X,'LEVEL NUMBER',4X,'SCALING FACTORS',4X,'THRESHOLDS',
1   4X,'EXPANSION FACTORS')
255 FORMAT(6X,I2,17X,F7.4,9X,F7.4,15X,F7.4)
TYPE *, 'DO YOU WANT TO CHANGE ANY OF THESE?(YES=1,NO=0)'.
ACCEPT *,1
IF(1.EQ.0)GO TO 955
TYPE *, 'THE FOLLOWING ARE THE CURRENT VALUES'
WRITE(5,111)MSF,KQ,B,LOEXP,HIEXP
111 FORMAT(/6X,'MSF=',F5.2/6X,'KQ=',I2/6X,'B=',F5.2/6X,'LOEXP=',F5.2
1   /6X,'HIEXP=',F5.2/)

```



```

GO TO 565
565 IF(ILEC-#)
565 IF(ISELEC.EQ.1)ITSF=IFILE
565 IF(ISELEC.EQ.2)ICSF=IFILE
565 IF(ISELEC.EQ.3)IOSF=IFILE
575 ISELEC=ISELEC+1
575 IF(ISELEC.EQ.2)IFILE=ICSF
575 IF(ISELEC.EQ.3)IFILE=IOSF
575 IF(ISELEC.LE.3)GO TO 545
GO TO 423
575 CONTINUE
585 DO 585 I=1,8
585 ANS3(I)=#
585 CONTINUE
TYPE * THE FOLLOWING INDICATE WHICH PLOTTABLE FILES WILL BE'
TYPE * OUTPUT BY THIS COMMAND FILE'
TYPE * LB=NOT OUTPUT, 1=WILL BE OUTPUT1'
TYPE 585
TYPE 595,ITFF
TYPE 595,ITSE
TYPE 605,ITCS
TYPE 605,ITPF
TYPE 615,IRFF
TYPE 615,IRSE
TYPE 625,IRCS
TYPE 625,IRPF
595 FORMAT(
595 FORMAT(
605 FORMAT(
605 FORMAT(
615 FORMAT(
615 FORMAT(
625 FORMAT(
625 FORMAT(
TYPE * TO CHANGE ANY OR ALL OF THESE TYPE=---'
TYPE * [SELECTION] (SELECTION) (SELECTION)...--UP TO 8 CHOICES'
ACCEPT 635,ANS3
535 FORMAT(8(11,1X))
CHK=#
DO 635 I=1,8
CHK=CHK+ANS3(I)
635 CONTINUE
IF(CHK.EQ.#)GO TO 685
IFILE=ITFF
ISELEC=1
645 DO 645 I=1,8
IF(ANS3(I).EQ.ISELEC)GO TO 655
645 CONTINUE
GO TO 675
655 IF(IFILE.NE.#)GO TO 655
IFILE=1
GO TO 665
1)TRANSMITTER FILTERED SPEECH SET=,11)
2)TRANSMITTER SIGNAL ENERGY SET=,11)
3)TRANSMITTER CLIPPED SPEECH SET=,11)
4)TRANSMITTER PITCH FILE SET=,11)
5)RECEIVER FILTERED SPEECH SET=,11)
6)RECEIVER SIGNAL ENERGY SET=,11)
7)RECEIVER CLIPPED SPEECH SET=,11)
8)RECEIVER PITCH FILE SET=,11)

```

```

665 IF(ISELEC.EQ.1)ITPF=IFILE
666 IF(ISELEC.EQ.2)ITSE=IFILE
IF(ISELEC.EQ.3)ITCS=IFILE
IF(ISELEC.EQ.4)ITPF=IFILE
IF(ISELEC.EQ.5)IRPF=IFILE
IF(ISELEC.EQ.6)IRSE=IFILE
IF(ISELEC.EQ.7)IRCS=IFILE
IF(ISELEC.EQ.8)IRPF=IFILE
675 ISELEC=ISELEC+1
IF(ISELEC.EQ.2)IFILE=ITSE
IF(ISELEC.EQ.3)IFILE=ITCS
IF(ISELEC.EQ.4)IFILE=ITPF
IF(ISELEC.EQ.5)IFILE=IRPF
IF(ISELEC.EQ.6)IFILE=IRSE
IF(ISELEC.EQ.7)IFILE=IRCS
IF(ISELEC.EQ.8)IFILE=IRPF
IF(ISELEC.LE.8)GO TO 645
GO TO 585
685 CONTINUE
C*****
C*****CHOICE OF FILTERS.
C*****
TYPE ** DO YOU WANT TO FILTER THE PITCH SPEECH?
TYPE ** [YES=1,NO=5]
ACCEPT **NOFLT
IF(NOFLT.EQ.5)GO TO 755
TYPE ** WHICH FILTER DO YOU WANT?
TYPE ** 1)8KHZ INPUT SPEECH WITH FILTER CUTOFF FREQ=555HZ.
TYPE ** 2)8KHZ INPUT SPEECH WITH FILTER CUTOFF FREQ=1KHZ.
TYPE ** 3)6.4KHZ INPUT SPEECH WITH FILTER CUTOFF FREQ=555HZ.
TYPE ** 4)6.4KHZ INPUT SPEECH WITH FILTER CUTOFF FREQ=1KHZ.
ACCEPT **1
DO 565 J=1,5
HF(J)=H(J,1)
565 CONTINUE
GO TO 715
755 DO 715 J=1,5
HF(J)=5
715 CONTINUE
C*****
C*****CHOICE OF PROGRAM PATHS.
C*****
TYPE ** WHICH PROGRAM PATH DO YOU WANT TO TAKE?
TYPE ** 1)TDHS-ARC WITH PITCH PASSED TO THE RCVR.
TYPE ** 2)TDHS-ARC WITH PITCH INDEPENDENTLY GENERATED.
TYPE ** 3)TDHS ONLY WITH PITCH PASSED TO THE TDHE ALGORITHM.
TYPE **

```

```

TYPE *.:
TYPE *.:
4)TDMS ONLY WITH PITCH INDEPENDENTLY GENERATED
IN THE TDHE ALGORITHM.
ACCEPT *,IPATH
C*****
C*****
C*****
C*****
C*****CHOICE OF INITIAL PREDICTOR COEFFICIENTS AND GAIN
C*****
84# TYPE *.: THE FOLLOWING ARE THE INITIAL PREDICTOR
TYPE *.: COEFFICIENTS AND PREDICTOR GAIN.
DO 8# I=1,N
TYPE 81# I,AB(I)
8# CONTINUE
TYPE 82# GAIN
81# FORMAT( ' B(' ,I2,' )=' ,2X,F8.4)
82# FORMAT( ' GAIN=' ,2X,F8.4)
TYPE *.: DO YOU WANT TO ENTER NEW VALUES?
TYPE *.: YES=1 NO=0
ACCEPT *,NAI
IF(NAI.EQ.0)GO TO 83#
TYPE *.: ENTER THE NEW INITIAL PREDICTOR COEFFICIENTS
ACCEPT *,(AB(I),I=1,N)
TYPE *.: ENTER THE NEW GAIN FACTOR
ACCEPT *,GAIN
GO TO 84#
C*****
C*****
C*****
C*****
C*****OUTPUT BLOCK.
C*****
83# WRITE(3,9)(FNAME(J),J=1,16)
WRITE(3,*)NBL
WRITE(3,*)CLPP
WRITE(3,*)KBLK1,ITMIN,ITMAX
WRITE(3,*)ALP,ALAD,ABIAS,G.M,RMSHIN,SMIN
WRITE(3,*)KQ
WRITE(3,*)(T(I),I=1,NQQQ)
WRITE(3,*)(OUT(I),I=1,NQQQ)
WRITE(3,*)(EXPN(I),I=1,NQQQ)
WRITE(3,*)ALPHA
WRITE(3,*)ITCS,ITPF,ITSE,ITSF,IRCS,IRPF,IRSE,ICSF,IOSF,ITFF,IRFF
WRITE(3,*)NOFLT
WRITE(3,*)(HF(I),I=1,5)
WRITE(3,*)IPATH
WRITE(3,*)(AB(I),I=1,N)
WRITE(3,*)GAIN
CLOSE(UNIT=3)
STOP
END
C*****
C*****
C*****

```

# THE

**AL3-3911:3T111**

BY: JAMES D. MILLS, 18 APR 1983

DESCRIPTIVE TITLE: 16KB/S TRANSMITTER SIMULATION FOR  
NON-FINITE WORD LENGTH ALGORITHM.

DESCRIPTION: THIS PROGRAM USES THE OUTPUT PARAMETER FILE OF P16F.FTN, RENAMED CMDPARA.DAT, TO SET ALL THE OPTIONS OF THE TRANSMITTER. FOR THIS REASON, THE OPERATOR IS NOT PROMPTED FOR ANY INFORMATION DURING A RUN. THIS PROGRAM REPRESENTS THE NON-FINITE WORD LENGTH SIMULATION OF THE TRANSMITTER FOR THE TDHS-ARC ALGORITHM. THE TRANSMISSION RATE IS ASSUMED TO BE 16KB/S.

```

C
C
      INTEGER HD(40), IPICH(40), FNAME(16), SQ, Q, VARE6
      INTEGER*2 CLPSP(512), STAT
      INTEGER*4 STRT, BUFSZ
      DIMENSION H(40), SQ(16), JSBUF(512), IFSP(512), IVY(16)
      REAL ALPHA
C*****
C*****
C*****
      COMMON /PRED/G, RMSHIN, ALP, ALAD, ABIAS, A(12), VHAT(12), EP, EV
      1      .SPERB1(20), SPERB2(20), V(12), SNRB(20), SNKOB(20), B(12), NSPSAM
      2      .ISTAT1(40), N, KQ
      COMMON /RMSB/RMS, ENGY1, ENGY2, ENGY3, ENGY4, SPER1, SPER2, NBL, IARG
      COMMON /CLIPP/CLPP, ALPHA, ISEC
      COMMON /ADDN/STAT(40)
      COMMON /VARE6/VARE6
      COMMON /IFILE/ITCS, ITPF, ITSE, ITSF, IRCS, IRPF, IRSE, ICSF, IOSF, ITFF, IRFF
      1      .IPATH
      COMMON /FILT/HF(5), IX(9), IFSC, NOFLT
C*****
C*****
C*****
      MODN(K) = K - (K-1)/16 * 16
C*****
C*****
C*****
C*****
C*****
C*****
C*****
      OPEN(UNIT=8, TYPE='OLD', NAME='CHD PARA.DAT')      IOPEN THE OPTION FILE.
C*****
C*****
C*****
      OPEN THE SPEECH FILE; READ THE HEADER AND SPEECH.
C*****
      READ(8,10) FNAME
      10  FORMAT(16A2)
      OPEN(UNIT=1, TYPE='OLD', READONLY, NAME=FNAME, SHARED); OPEN SPEECH FILE
      READ(1,20) NSENT, IRATE, NSAMP, IUPPR, ILOWR, NTERMS, HD
      20  FORMAT(16I5, 10X, 40A1)
      30  FORMAT(16I5)
      40  CONTINUE
C*****
C*****
C*****
      CREATE NEW FILE FOR QUANTIZER OUTPUT. WRITE
      THE HEADER.
C*****
C*****
      OPEN(UNIT=2, TYPE='NEW', NAME='QUANT.DAT', CARRIAGECONTROL='LIST')
      50  FORMAT(40A1)
      NSPSAM = NSAMP/2
      WRITE(2,20) NSENT, IRATE, NSPSAM, IUPPR, ILOWR, NTERMS, HD
C*****
C*****
C*****
      DEFINE AND INITIALIZE VARIOUS PARAMETERS.
C*****
      K1=773

```

```

K2=119
DO 68 JJ=1,188
  68 CALL RANDU(K1,K2,XX)
  NUMP = 8
  NADD = 8
  ISEC=8
  IFSC=8
  READ(8,*)NBL
  READ(8,*)CLPP
  READ(8,*)KBLK1,ITMIN,ITMAX
C*****
C*****INITIALIZATION SUBROUTINE.
C*****
  CALL INSTRT
C*****
C*****FILE MANAGEMENT FOR RUNS.
C*****
  IF(ITCS.NE.1)GO TO 188
  OPEN(UNIT=18,TYPE='NEW',NAME='TCLPSP.DAT',CARRIAGECONTROL='LIST')
  WRITE(18,28)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
  188 IF(ITSE.NE.1)GO TO 288
  OPEN(UNIT=7,TYPE='NEW',NAME='TSENGY.DAT',CARRIAGECONTROL='LIST')
  WRITE(7,28)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
  288 IF(ITPF.NE.1.AND.IPATH.NE.1.AND.IPATH.NE.3)GO TO 388
  OPEN(UNIT=4,TYPE='NEW',NAME='TPITCH.DAT',CARRIAGECONTROL='LIST')
  WRITE(4,28)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
  388 IF(ITFF.NE.1)GO TO 488
  OPEN(UNIT=11,TYPE='NEW',NAME='TFILTER.DAT',CARRIAGECONTROL='LIST')
  WRITE(11,28)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
  488 IF(IPATH.NE.3.AND.IPATH.NE.4)GO TO 688
  OPEN(UNIT=12,TYPE='NEW',NAME='TZHAT.DAT',CARRIAGECONTROL='LIST')
  WRITE(12,28)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
  688 CONTINUE
C*****
C*****
C*****
C*****BEGIN MAIN LOOP. SPEECH BUFFER CREATED AND CLIPPED
C*****SPEECH GENERATED BY SUBROUTINE.
C*****
  SENGY=8.
  BUF SZ=2*ITMAX
  BUF SZ=(BUF SZ/16)*16+16
  READ(1,38)(ISBUF(1),1=1,BUF SZ)
  STRT=1
  22 CONTINUE
  IF(NOF LT.EQ.1)GO TO 498
  DO 585 I=STRT,BUF SZ
    IFSP(I)=ISBUF(I)
  585 CONTINUE
  GO TO 485
C*****
C*****LOW PASS FILTER SUBROUTINE.

```

```

C*****
498 CALL IFILT(ISBUF,STRT,BUFSZ,IFSP)
C*****
C*****SUBROUTINE TO CLIP SPEECH.
C*****
485 CALL CLP(IFSP,STRT,BUFSZ,CLPSP,SENGY)
C*****
C*****PITCH DETERMINED BY AUTOCORRELATION.
C*****
21 IBIGG=-1555
I2SUM=-1555
DO 12 IT=ITMIN,ITMAX+1
  ISUM=0
  DO 13 J=1,KBLK1
    IF(((CLPSP(J+IT).LT.B).AND.(CLPSP(J).LT.B)).OR.
      ((CLPSP(J+IT).GT.B).AND.(CLPSP(J).GT.B)))
      ISUM=ISUM+1
    IF(((CLPSP(J+IT).LT.B).AND.(CLPSP(J).GT.B)).OR.
      ((CLPSP(J+IT).GT.B).AND.(CLPSP(J).LT.B)))
      ISUM=ISUM-1
  13 CONTINUE
  IF(I2SUM.GT.IBIGG.AND.I2SUM.GT.ISUM)NP=IT-1
  IF(I2SUM.GT.IBIGG.AND.I2SUM.GT.ISUM)IBIGG=I2SUM
  I2SUM=ISUM
12 CONTINUE
  NUMP=NUMP+1
C*****
C*****RANDOM PITCH GENERATOR.
C*****
NINC=MODN(NUMP)
IF(NP.GT.ITMIN) GO TO 29
CALL RANDU(K1,K2,XX)
NP=XX*(ITMAX-ITMIN)/2+ITMIN
WRITE(5,462)NUMP,NP
29 IPICH(NINC)=NP
462 FORMAT(/6X,'NUMP=',I4,6X,'NP IS=',I4)
IF(ITPF.NE.1.AND.IPATH.NE.1.AND.IPATH.NE.3)GO TO 555
IF(NINC.EQ.16)WRITE(4,38)(IPICH(J),J=1,16)
C*****
C*****WINDOW FORMED BY SUBROUTINE FOR SPEECH COMPRESSION
C*****FROM PITCH GENERATED.
C*****
555 CALL WINDOW(N,NP)
C*****
C*****SPEECH IS COMPRESSED AND ARC IS EMPLOYED.
C*****
DO 14 I=1,NP
  NUM4=NP+I

```

```

VV=FLOAT(ISBUF(NUM4))+H(I)*FLOAT(ISBUF(I))-ISBUF(NUM4))
IARG=NADD+I
N1=MODN(IARG)
VV=VV
IF(VV.GT.B.)VV=VV+B.5
IF(VV.LT.B.)VV=VV-B.5
IF(VV.GT.2B47.)VV=2B47.B
IF(VV.LT.-2B48.)VV=-2B48.B
IVV(N1)=IFX(VV)
IF((IPATH.EQ.3.OR.IPATH.EQ.4).AND.N1.EQ.16)WRITE(12,3B)
1      (IVV(J),J=1,16)
      CALL ARC(VV,Q,VHAT)
      JQ=Q
      STAT(JQ)=STAT(JQ)+1
      SQ(N1)=Q
      IF(N1.EQ.16)WRITE(2,33)SQ
14 CONTINUE
C*****
C*****UPDATE FOR NEXT BLOCK FO SPEECH. BUFFERS ARE
C*****SHIFTED OVER 2*NP PLACES AND THEN FILLED AGAIN.
C*****IF THE END OF THE FILE IS REACHED, ZEROS ARE AP-
C*****PENDDED TO FINISH PROCESSING THE FILE.
C*****
      NADD=NADD+NP
      NP2=2*NP
      INBUFF=BUF SZ-NP2
      DO 15 I=1,INBUFF
        ISBUF(I)=ISBUF(NP2+I)
        CLPSP(I)=CLPSP(NP2+I)
        IFSP(I)=IFSP(NP2+I)
15 CONTINUE
      BUF SZ=2*ITMAX
      MORE=BUF SZ-INBUFF
      IF(MOD(MORE,16).EQ.B)GO TO 25
      MORE=(MORE/16)*16+16
25 B:FSZ=INBUFF+MORE
      LEFT=NSAMP-2*NADD
      IF(MORE.GT.NLEFT)GO TO 23
      READ(1,3B,END=26)((ISBUF(INBUFF+I),I=1,MORE)
26 STRT=INBUFF+1
      GO TO 22
23 CHK=B.
      DO 17 I=1,INBUFF
        CHK=CHK+FLOAT(ABS(ISBUF(I)))
17 CONTINUE
      IF(CHK.EQ.B.)GO TO 24
      NZEROS=MORE-NLEFT
      READ(1,3B,END=27)((ISBUF(INBUFF+I),I=1,NLEFT)
27 DO 16 I=1,NZEROS
        ISBUF(INBUFF+NLEFT+I)=B

```

```

16 CONTINUE
   STRT=INBUFF+1
   GO TO 22
24 NONES=16-N1
   DO 18 I=1,NONES
      SO(N1+I)=1
18 CONTINUE
   WRITE(2,33)SO
   IF(ITPF.EQ.1.OR.IPATH.EQ.1.OR.IPATH.EQ.3)WRITE(4,38)
1   (IPICH(J),J=1,NINC)
   IF(IPATH.EQ.3)WRITE(12,38)(IVV(J),J=1,N1)
C*****STATISTICS GENERATED FOR OUTPUT FILE BY SUBROUTINE.
C*****
   CALL INEND
C*****
C*****CLOSE FILES.
C*****
33 FORMAT(16I2)
   CLOSE(UNIT=7)
   CLOSE(UNIT=1)
   CLOSE(UNIT=2)
   CLOSE(UNIT=8)
   CLOSE(UNIT=4)
   CLOSE(UNIT=11)
   STOP
   END

```



129

```

*****
C*****
C*****THE VALUE OF THE QUANTIZED ERROR IS DETERMINED.
C*****
C*****
C*****Y=2.*F*OUT(J)*SIZE
C*****
C*****
C*****THE STANDARD DEVIATION ESTIMATE IS UPDATED.
C*****
C*****
C*****SIZE=EXPN(J)*SIZE
C*****SIZE=AMAX1(SIZE,RMS)
C*****RETURN
C*****END

```





```

*****
C*****
C*****
C*****ENERGY VALUES FOR STATISTICS ARE INITIALLY SET TO
C*****ZERO.
C*****

```

```

EV=0.
EP=0.
ENGY1=0.
ENGY2=0.
J1=0
ENGY3=0.
ENGY4=0.
ENGY5=0.
ENGY6=0.
ENGY7=0.
ENGY8=0.
NDIV=0
SPER1=0.
SPER2=0.
DO 621 I=1,KO
621 ISTAT(I)=0
END

```



```

C*****ERROR SAMPLE GENERATED.
C*****
C*****
C*****ERROR-Y-PRE
C*****ERROR1=Y-PRE1
C*****QUANTIZER SUBROUTINE IS ENGAGED.
C*****
C*****CALL QUANT(ERROR,EQ,IOUT)
C*****
C*****UPDATE PREDICTOR VALUES.
C*****
C*****ISTAT1(IOUT)=ISTAT1(IOUT)+1
C*****Q=IOUT
C*****DO 125 I=1,N
C*****J=N+2-I
C*****V(J)=V(J-1)
C*****125 VJAT(J)=VJAT(J-1)
C*****VJAT(1)=PRE+EQ
C*****V(1)=V
C*****VJAT=VJAT(1)
C*****
C*****ADAPTATION OF PREDICTOR COEFFICIENTS IS ACCOMPLISHED
C*****BY A FIXED PERCENTAGE COMPUTATION.
C*****
C*****IF(G.EQ.B.B)GO TO 134
C*****DO 133 I=1,N
C*****A(I)=A(I)-B(I)
C*****133 CONTINUE
C*****DO 135 I=1,N
C*****SVL=SIGN(1.,EQ)*SIGN(1.,VJAT(I+1))
C*****IF((EQ.EQ.B..OR.VJAT(I+1).EQ.B..).AND.A(I).GE.B.)SVL=-1.
C*****IF((EQ.EQ.B..OR.VJAT(I+1).EQ.B..).AND.A(I).LT.B.)SVL=+1.
C*****AINT=SVL*AMAX1(ABS(A(I))*G,ABIAS)
C*****A(I)=ALAD*(A(I)+AINT)
C*****135 CONTINUE
C*****DO 155 I=1,N
C*****A(I)=A(I)+B(I)
C*****155 CONTINUE
C*****
C*****ENERGY STATISTICS ARE GENERATED FOR DETERMINATION
C*****OF SNR VALUES REPORTED IN STATISTICS FILE.
C*****
C*****134 IRM=MOD(IARG,NBL)
C*****ENG1=ENG1+Y**2
C*****ENG2=ENG2+(ERROR-EQ)**2
C*****ENG3=ENG3+ERROR**2
C*****ENG4=ENG4+ERROR1**2

```

```

ENG5=ENG5+Y*YHAT
ENG6=ENG6+YHAT**2
ENG7=ENG7+YHAT
ENG8=ENG8+Y
NDIV=NDIV+1
IF(IRM .NE. 8)GOTO 133
J1=J1+1
EV=EV+ENG1
EP=EP+ENG2
IF(ENG3.NE.8..AND.ENG1.NE.8.)SPER1(J1)=18.*ALOG18(ENG1/ENG3)
IF(ENG4.NE.8..AND.ENG1.NE.8.)SPER2(J1)=18.*ALOG18(ENG1/ENG4)
IF(ENG2.NE.8..AND.ENG1.NE.8.)SNR1(J1)=18.*ALOG18(ENG1/ENG2)
IF(ENG2.NE.8..AND.ENG3.NE.8.)SNR2(J1)=18.*ALOG18(ENG3/ENG2)
SPER1=SPER1+ENG3
SPER2=SPER2+ENG4
ENG1=8.
ENG2=8.
ENG3=8.
ENG4=8.
CONTINUE
RETURN
END

```

133



```

ARGI=ISUM
SUM=ALOG(ARG1)
DO 508 I=1,KQ
  ARG=ISTAT1(I)*.5.501
  SUM=SUM-(ARG*ALOG(ARG))/ARG1
  BITS=SUM/ALOG(2.)
  WRITE(WARE6,402)SNR,BITS,(ISTAT1(I),I=1,KQ)
402 FORMAT(6X,'SNR INLOOP=',F7.3X,'H=',F4.2X,'OP',
1 1015,3/(22X,1015))
  IF(WARE6.EQ.6)WRITE(5,402)SNR,BITS,(ISTAT1(I),I=1,KQ)
  WRITE(WARE6,404)
  FORMAT(///6X,'SAMPLE NUMBER',3X,'SNR',7X,'SPER',5X,
1 1015,3/(22X,1015))
  NB=ISUM/NBL
DO 408 I=1,NB
  IS=(I-1)*NBL+1
  IE=IS+NBL-1
  WRITE(WARE6,412)IS,IE,SNRB(I),SPERB1(I),SPERB2(I),SNROB(I)
408 CONTINUE
  FORMAT(6X,15,'-',15,6X,F7.2X,F7.2X,F7.2X,F7.2X,F7.2X,F7.2)
  WRITE(WARE6,416)SPER,SPER1,SNRO
416 FORMAT(//6X,'PREDICTOR PERFORMANCE=',F8.2/6X,
1 1015,3/(22X,1015))
  2 1015,3/(22X,1015)
  RATIO='F8.2'
  NQUA=0
DO 418 I=1,32
  NQUA=NQUA+STAT(I)
418 DO 420 I=1,32
  PROB(I)=FLOAT(STAT(I))/FLOAT(NQUA)
  CONTINUE
  WRITE(WARE6,422)
  FORMAT(//6X,'LEVEL NUMBER',6X,'PROBABILITY',6X,'FREQUENCY',/)
424 FORMAT(9X,12,12X,F7.4,10X,15)
DO 426 I=1,32
  WRITE(WARE6,424)I,PROB(I),STAT(I)
426 CONTINUE
  WRITE(WARE6,505)(NQ+1),T(NQ+2)
505 FORMAT('THE NUMBER OF TIMES THE THRESHOLD VALUE OF ',F7.4,'WAS'
1 1015,3/(22X,1015))
  DEN1=ENG7
  DEN2=(ENG7**2)-FLOAT(NDIV)*ENG6
  IF(DEN1.EQ.0.)TYPE='BETA DENOM = 0. CANNOT DIVIDE'
  IF(DEN2.EQ.0.)TYPE='K DENOM = 0. CANNOT DIVIDE'
  IF(DEN1.EQ.0. OR DEN2.EQ.0.)GO TO 700
  TOP1=ENG6*ENG8*ENG7-FLOAT(NDIV)*ENG5*ENG6
  BETA=(ENG5/DEN1)-(TOP1/(DEN1*DEN2))
  TOP2=ENG8*ENG7-FLOAT(NDIV)*ENG5
  SNRKAY=TOP2/DEN2
  SNRI=EV-2.*SNRKAY*ENG5-2.*BETA*ENG8+2.*BETA*SNRKAY*ENG7
  SNR2=(BETA**2)*FLOAT(NDIV)+(SNRKAY**2)*ENG6
  DENSNR=SNRI+SNR2
  IF(DENSNR.EQ.0.)TYPE='NOISE OF MAXIMIZED SNR = 0.'

```

```

IF(DENSNR.EQ.B.)GO TO 755
SNRMAX=15.*ALOG10(EV/DENSNR)
WRITE(VAREG,755)SNRKAY,BETA,SNRMAX
755 FORMAT(' THE SNR SCALING FACTOR IS K=','F10.5/',' THE SNR DC',
1      ' , OFFSET IS BETA =','F10.5/',' AND THE MAXIMIZED SNR IS =','F10.5)
755 RETURN
END

```



```

CLM=-CLPLVL
XFLOT=FLOAT(IFSP(1))
CLPSP(1)=0
IF (XFLOT.GT.CLPLVL)CLPSP(1)=+1
IF (XFLOT.LT.CLPLVL)CLPSP(1)=-1
CLIPFI(MSEC)=CLPSP(1)
IF (ITCS.NE.1)GO TO 21
IF (MSEC.EQ.16)WRITE(10,10)(CLIPFI(JJ),JJ=1,16)

21 CONTINUE
10 FORMAT(16I5)
RETURN
END

```



**555 CONTINUE  
RETURN  
END**

မမမမမမ

[illegible]

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 104

**BY: JAMES D. MILLS, 18 APR 1983**

DESCRIPTION: THIS PROGRAM USES THE OUTPUT PARAMETER FILE OF P16F.FTN, RENAMED CMDPARA.DAT, TO SET ALL THE OPTIONS OF THE RECEIVER. FOR THIS REASON, THE OPERATOR IS NOT PROMPTED FOR ANY INFORMATION DURING A RUN. THIS PROGRAM REPRESENTS THE NON-FINITE WORD LENGTH SIMULATION OF THE RECEIVER FOR THE TDHS-ARC ALGORITHM. THE TRANSMISSION RATE IS ASSUMED TO BE 16KB/S.

[illegible]



```

C*****
C*****
C*****OPEN FILES AND WRITE HEADERS.
C*****
      OPEN(UNIT=1,TYPE='OLD',NAME='DECO.DAT')
      READ(1,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
      NSPSAM=NSAMP+NSAMP
      IF(1CSF.NE.1)GO TO 400
      OPEN(UNIT=3,TYPE='NEW',NAME='ZHAT.DAT',CARRIAGECONTROL='LIST')
      WRITE(3,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
400  IF(1OSF.NE.1)GO TO 410
      OPEN(UNIT=7,TYPE='NEW',NAME='SHAT.DAT',CARRIAGECONTROL='LIST')
      WRITE(7,20)NSENT,IRATE,NSPSAM,IUPPR,ILOWR,NTERMS,HD
410  IF(1RSE.NE.1)GO TO 420
      OPEN(UNIT=9,TYPE='NEW',NAME='RSENGY.DAT',CARRIAGECONTROL='LIST')
      WRITE(9,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
420  IF(1RPF.NE.1)GO TO 430
      OPEN(UNIT=2,TYPE='NEW',NAME='RPITCH.DAT',CARRIAGECONTROL='LIST')
      WRITE(2,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
430  IF(1RCS.NE.1)GO TO 440
      OPEN(UNIT=10,TYPE='NEW',NAME='RCLPSP.DAT',CARRIAGECONTROL='LIST')
      WRITE(10,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
440  IF(1RFF.NE.1)GO TO 445
      OPEN(UNIT=11,TYPE='NEW',NAME='RFILTER.DAT',CARRIAGECONTROL='LIST')
      WRITE(11,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
445  IF(1PATH.EQ.1.OR.1PATH.EQ.2)GO TO 435
      OPEN(UNIT=12,TYPE='OLD',NAME='TZHAT.DAT')
      READ(12,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
435  IF(1PATH.EQ.2.OR.1PATH.EQ.4)GO TO 425
      OPEN(UNIT=13,TYPE='OLD',NAME='TPITCH.DAT')
      READ(13,20)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
425  CONTINUE
C*****
C*****
C*****INITIALIZE REMAINING PARAMETERS.
C*****
      ISEC=0
      IFSC=0
      JJ=0
      NLEFT=0
      INC1=0
      INC2=0
      PI=0
      NADD=0
      SENGY=0.
      K1=773
      K2=119
      DO 450 JQ=1,100
450  CALL RANDU(K1,K2,XX)
C*****
C*****

```

```

C*****
C*****SET UP MAIN LOOP INITIAL SPEECH BLOCKS AND
C*****VARIABLES.
C*****
      BUFSZ=2*ITMAX
      BUFSZ=MODN(BUFSZ)
      DO 485 I=1,ITMAX
         QCOMP(I)=5.
485 CONTINUE
      IF(IPATH.EQ.3)READ(12,55)(Q(J),J=1,BUFSZ)
      IF(IPATH.LE.2)READ(1,45)(Q(J),J=1,BUFSZ)
      NNEED=BUFSZ
      INC3=ITMAX
      ISTRTI=1
      IFNSHI=NNEED+ITMAX
      NSTOP=BUFSZ+ITMAX+16
C*****
C*****
C*****BEGIN MAIN LOOP. INVERSE ARC ENGAGED FOR EACH
C*****SAMPLE AND RESULTING BLOCKS OF SPEECH ARE
C*****EXPANDED USING TDHS.
C*****
      495 DO 555 I=1,NNEED
         IF(IPATH.LE.2)CALL ARCR(Q(I),VY)
         IF(IPATH.EQ.3)VY=Q(I)
         INC2=INC2+1
         INC1=MODN(INC2)
         QCOMP(INC3+1)=VY
         IF(VY.GT.5)VY=VY+5.5
         IF(VY.LT.5)VY=VY-5.5
         IF(VY.GT.2547)VY=2547.5
         IF(VY.LT.-2548)VY=-2548.5
         IZMAT(INC1)=IFIX(VY)
         IF(ICSF.NE.1)GO TO 555
         IF(INC1.EQ.16)WRITE(3,55)(IZMAT(K),K=1,16)
555 CONTINUE
      IF(IPATH.EQ.1.OR.IPATH.EQ.3)GO TO 355
C*****
C*****COMPRESSED SPEECH SI FILTERED BY SUBROUTINE.
C*****
      IF(NOFLT.EQ.1)GO TO 495
      DO 505 I=ISTRTI,IFNSHI
         FS(I)=QCOMP(I)
505 CONTINUE
      GO TO 485
495 CALL IFLT(QCOMP,ISTRTI,IFNSHI,FS)
C*****
C*****CLIP THE COMPRESSED SPEECH BY SUBROUTINE.
C*****
      485 CALL CLPPP(FS,ISTRTI,IFNSHI,CLPSP,SENGY)
C*****

```

```

C*****
C*****PITCH DETERMINED BY AUTOCORRELATION.
C*****
515 IBIGG=-1555
12SUM=-1555
DO 525 IT=ITMIN,ITMAX+1
    ISUM=0
    IBGN=ITMAX
    ISTP=3-KBLK1
    DO 525 J=IBGN,ISTP
        IF(((CLPSP(J-IT).LT.0).AND.(CLPSP(J).GT.0)).OR.
            ((CLPSP(J-IT).GT.0).AND.(CLPSP(J).GT.0)))
            ISUM=ISUM+1
        IF(((CLPSP(J-IT).LT.0).AND.(CLPSP(J).GT.0)).OR.
            ((CLPSP(J-IT).GT.0).AND.(CLPSP(J).LT.0)))
            ISUM=ISUM-1
    CONTINUE
    IF(12SUM.GT.IBIGG.AND.12SUM.GT.ISUM)NP=IT-1
    IF(12SUM.GT.IBIGG.AND.12SUM.GT.ISUM)IBIGG=12SUM
    12SUM=ISUM
525 CONTINUE
355 JJ=JJ+1
C*****
C*****RANDOM PITCH GENERATOR.
C*****
JINC=MODN(JJ)
IF(IPATH.EQ.2.OR.IPATH.EQ.4)GO TO 355
IF(JINC.EQ.1)READ(13,55,END=365)((IPICH(J),J=1,16)
355 NP=IPICH(JINC)
GO TO 515
365 NP=ITMIN
355 IF(NP.GT.ITMIN)GO TO 515
CALL RANDU(K1,K2,XX)
NP=XX*(ITMAX-ITMIN)/2+ITMIN
WRITE(5,288)JJ,NP
515 IPICH(JINC)=NP
288 FORMAT(6X,'NUMP=',14,6X,'NP IS=',14,12X,'RANDOM PITCHIIIIII')
IF(IRPF.NE.1)GO TO 535
IF(JINC.EQ.16)WRITE(2,55)((IPICH(K),K=1,16)
535 NP2=NP+NP
C*****
C*****WINDOW FORMED FOR SPEECH EXPANSION FROM
C*****PITCH GENERATED.
C*****
DO 545 I=1,NP2
    H(I)=1.-FLOAT(I-1)/FLOAT(NP2-1)
545 CONTINUE
C*****
C*****SPEECH IS EXPANDED.
C*****

```

```

C*****
555 P1=ITMAX-NP
NP3=NP2+NP
DO 555 I=1,NP3
  VYAT(I)=OCOMP(P1+I)
555 CONTINUE
DO 565 I=1,NP2
  SHAT(I)=VYAT(I)+H(I)*(VYAT(I+NP)-VYAT(I))
  IARG=NADD+I
  N1=MODN(IARG)
  XX=SHAT(I)
  IF(XX.GT.5.5)XX=XX+5.5
  IF(XX.LT.5.5)XX=XX-5.5
  IF(XX.GT.2547.5)XX=2547.5
  IF(XX.LT.-2548.5)XX=-2548.5
  SQ(N1)=FIX(XX)
  IF(IOSF.NE.1)GO TO 565
  IF(N1.EQ.16)WRITE(7,55)(SQ(J),J=1,16)
565 CONTINUE
C*****
C*****SHIFT ALL VECTORS APPROPRIATELY BASED ON PITCH
C*****VALUE USED TO EXPAND SPEECH. SET UP VECTORS
C*****AND PARAMETERS FOR NEXT RUN THROUGH MAIN LOOP.
C*****
  NADD=NADD+NP+NP
  INC3=ITMAX-NP+NNEED+NLEFT
  ISTAT1=INC3+1
  NLEFT=NNEED-NP+NLEFT
DO 565 I=1,INC3
  OCOMP(I)=OCOMP(I+NP)
  CLPSP(I)=CLPSP(I+NP)
  FS(I)=FS(I+NP)
565 CONTINUE
  NNEED=MODN(NP)
  NTOT=INC3+NNEED
  IF(NTOT.GT.NSTOP)NNEED=NNEED-16
  IF(N1=INC3+NNEED
  IF(NADD.GE.NSPSAM)GO TO 655
  IF(IPATH.GE.3)READ(12,55,END=585)(Q(I),I=1,NNEED)
  IF(IPATH.LE.2)READ(1,45,END=585)(Q(I),I=1,NNEED)
  GO TO 595
585 DO 585 J=1,NNEED
  Q(I)=5
585 CONTINUE
595 GO TO 495
C*****
C*****AT END OF FILE, FINISH WRITING VALUES TO APPROPRIATE
C*****FILES.
C*****
  655 IF(IOSF.NE.1)GO TO 615

```

```

WRITE(7,55)(SQ(J),J=1,M1)
61# IF(IRSE.NE.1)GO TO 62#
WRITE(9,55)(SENG(J),J=1,INC1)
62# IF(ICSF.NE.1)GO TO 63#
WRITE(3,55)(IZMAT(J),J=1,INC1)
63# IF(IRPF.NE.1)GO TO 64#
WRITE(2,55)(IPICH(J),J=1,JINC)
64# IF(IRCS.NE.1)GO TO 65#
WRITE(10,55)(CLPSP(J),J=1,MSEC)
65# IF(IRFF.NE.1)GO TO 67#
DO 66# I=1,IFINC
F=FS(I)
IF(F.GT.B.)F=F+B.B
IF(F.LT.B.)F=F-B.B
IFSP(I)=FIX(F)
66# CONTINUE
WRITE(11,55)(IFSP(J),J=1,IFINC)
C*****
C*****
C*****CLOSE FILES.
C*****
67# CLOSE(UNIT=1)
CLOSE(UNIT=3)
CLOSE(UNIT=4)
CLOSE(UNIT=7)
CLOSE(UNIT=2)
CLOSE(UNIT=11)
CLOSE(UNIT=10)
STOP
END

```





```

      READ(4,*)(OUT(I),I=1,NQQQ)
      READ(4,*)(EXPN(I),I=1,NQQQ)
      READ(4,*)ALPHA
C*****FILE MANAGEMENT FLAGS.
      READ(4,*)ITCS,ITPF,ITSE,ITSF,IRCS,IRPF,IRSE,ICSF,IOSF,ITFF,IRFF
      READ(4,*)NOFLT
      READ(4,*)(HF(I),I=1,5)
      READ(4,*)IPATH
      READ(4,*)(B(I),I=1,N)
      READ(4,*)GAIN
      SIZE=100.
      DO 118 I=1,12
        VHA(I)=0.
        A(I)=B(I)
118 CONTINUE
      DO 119 I=1,9
        XXX(I)=0
119 CONTINUE
      DO 120 I=1,NQQQ
        T(I)=T(I)*SMIN
        OUT(I)=OUT(I)*SMIN
120 CONTINUE
      RMS=RMSMIN
      RETURN
      END

```



```

C*****
DO 125 I=1,N
J=N+2-I
125 VHA(I)=VHA(J-I)
VHA(1)=PRE+EQ
VHA(1)=VHA(1)
C*****
C*****ADAPTATION OF PREDICTOR COEFFICIENTS IS ACCOMPLISHED
C*****BY A FIXED PERCENTAGE COMPUTATION.
C*****
IF(G.EQ.0.0)GO TO 131
DO 130 I=1,N
A(I)=A(I)-B(I)
130 CONTINUE
DO 130 I=1,N
SVL=SIGN(1.,EQ)*SIGN(1.,VHA(I+1))
IF((EQ.EQ.0.0).OR.VHA(I+1).EQ.0.0).AND.A(I).GE.0.0)SVL=-1.
IF((EQ.EQ.0.0).OR.VHA(I+1).EQ.0.0).AND.A(I).LT.0.0)SVL=+1.
AINT=SVL*AMAX1(ABS(A(I))*G,ABIAS)
A(I)=ALAD*(A(I)+AINT)
130 CONTINUE
DO 150 I=1,N
A(I)=A(I)+B(I)
150 CONTINUE
131 RETURN
END

```



```

CLPSP(I)=8
IF(XFLOT.GT.CLPLVL)CLPSP(I)=+1
IF(XFLOT.LT.CLM)CLPSP(I)=-1
CLIPFI(MSEC)=CLPSP(I)
IF(IRCS.NE.1)GO TO 21
IF(MSEC.EQ.16)WRITE(16,16)(CLIPFI(JJ),JJ=1,16)
21 CONTINUE
16 FORMAT(16I5)
RETURN
END

```



**5.55 CONTINUE  
RETURN  
END**

မမမမမမ

[illegible]

**APPENDIX C**  
**SUPPORTING PROGRAMS**



```

WRITE(3,20)ISENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,ND
DO 55 I=1,16
    IYV(I)=0
55 CONTINUE
    V1=0.
    V2=0.
    NUMP=NSAMP/512
    LFTOVR=(FLOAT(NSAMP)/512.-FLOAT(NUMP))*16.
    IF(LFTOVR.GT.0.)LFTOVR=LFTOVR+0.5
    IF(LFTOVR.LT.0.)LFTOVR=LFTOVR-0.5
    ILFT=IFIX(LFTOVR)
    DO 755 K=1,NUMP
        READ(2,30,END=100)(IVEC(I),I=1,512)
100 ISTOP=512
200 DO 600 JJ=1,ISTOP
    ICT=ICNT+1
    ICT=MODN(ICT)
    V2=V1
    V1=FLOAT(IVEC(JJ))
    VN=V2*(-.4)+V1
    IF(VN.GT.0.)VN=VN+0.5
    IF(VN.LT.0.)VN=VN-0.5
    IYV(ICT)=IFIX(VN)
    IF(ICT.EQ.16)WRITE(3,30)(IYV(KIK),KIK=1,16)
600 CONTINUE
755 IF(ISTOP.EQ.ILFT)GO TO 900
755 CONTINUE
855 READ(2,30,END=800)(IVEC(I),I=1,ILFT)
855 ISTOP=ILFT
900 GO TO 200
900 CLOSE(UNIT=3)
900 CLOSE(UNIT=2)
900 CLOSE(UNIT=1)
STOP
END

```

CCCCC



```

C*****
C*****IMPLEMENTATION OF FILTER
C*****
DO 55 I=1,16
  55 CONTINUE
  V1=S.
  V2=S.
  NUMLP=NSAMP/512.
  LFTOVR=(FLOAT(NSAMP)/512.-FLOAT(NUMLP))*16.
  IF(LFTOVR.GT.S.)LFTOVR=LFTOVR-S.5
  IF(LFTOVR.LT.S.)LFTOVR=LFTOVR-S.5
  ILFT=ifix(LFTOVR)
  DO 755 K=1,NUMLP
    READ(2,35,END=155)(IVEC(I),I=1,512)
  155 ISTOP=512
  255 DO 555 JJ=1,ISTP
    ICNT=ICNT+1
    ICT=MODN(ICNT)
    V2=V1
    V1=FLOAT(IVEC(JJ))
    VN=S.4*V2+V1
    IF(VN.GT.S.)VN=VN-S.5
    IF(VN.LT.S.)VN=VN-S.5
    IVN(1CT)=ifix(VN)
    IF(1CT.EQ.16)WRITE(3,35)(IVN(KIK),KIK=1,16)
  555 CONTINUE
  IF(ISTP.EQ.1LFT)GO TO 955
  755 CONTINUE
  READ(2,35,END=855)(IVEC(I),I=1,1LFT)
  855 ISTOP=1LFT
  GO TO 255
  955 CLOSE(UNIT=3)
  STOP
  END

```

```

C
C
C
C*****

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 104

BY: JAMES D. MILLS, 18 APR 1983

[illegible]

**ICNT=8**

```

TZSUM=B.
ZSUM=B.
K=256
C*****
C*****
C*****
155 READ(4,25,END=555)(ITZ(J),J=1,K)
255 READ(3,25,END=655)(IZ(J),J=1,K)
355 DO 455 I=1,K
      ICNT=ICNT+1
      TZ=FLOAT(ITZ(I))
      Z=FLOAT(IZ(I))
      TZ2=TZ**2.
      Z2=(TZ-Z)**2.
      FLAG=15.
      IF(Z2.NE.B.)FLAG=TZ2/(Z2*2)
      IF(FLAG.LE.1.)WRITE(6,*)ICNT,TZ,Z
      TZSUM=TZSUM+TZ2
      ZSUM=ZSUM+Z2
455 CONTINUE
      IF(K.EQ.256)GO TO 155
      SNR=15.*ALOG10(TZSUM/ZSUM)
      WRITE(5,35)SNR
      GO TO 755
555 K=J
655 K=J
755 GO TO 355
      CLOSE(UNIT=3)
      CLOSE(UNIT=4)
      STOP
      END

```

```

C
C
C
C
C*****
C*****

```

**NSENT** = 1  
**IRATE** = 8888  
**IUPPR** = 4888  
**ILOWR** = 2688

```

NTERMS = 40
OPEN(UNIT=1, TYPE='NEW', NAME='TONE.DAT', CARRIAGECONTROL='LIST')
C*****
C*****ENTER PARAMETERS FOR TONE.
C*****
TYPE *, ' ENTER VALUES FOR SINE FREQ,AMPLITUDE, SPEECH FREQ, '
TYPE *, ' AND NUMBER OF SAMPLES'
ACCEPT *, NFREQ,XMAX,NSFREQ,NSAMP
PI = 4.*ATAN(1.)
WRITE(1,30)NSENT,IRATE,NSAMP,IUPPR,ILOWR,NTERMS,HD
FORMAT(6I5,10X,40A1)
30  THETA=PI*(2. - FLOAT(NFREQ)/FLOAT(NSFREQ))
C*****
C*****
C*****SAMPLE GENERATION.
C*****
DO 50 N = 0,NSAMP
  FN = FLOAT(N)
  H0 = XMAX*SIN(THETA*FN)
  XX=H0
  N1 = MODN(N+1)
  IX(N1) = XX
  IF(N1.EQ.256)WRITE(1,60)IX
  IF( N.EQ.NSAMP)WRITE(1,60)(IX(J),J=1,N1)
50  CONTINUE
60  FORMAT(16I5)
STOP
END
C
C
C
C*****
C*****

```

[illegible][illegible]

~~~~~

~~~~~

~~~~~

**◎ 俗文化語彙 · 一**

UU UU

**MODN(K)=K-**

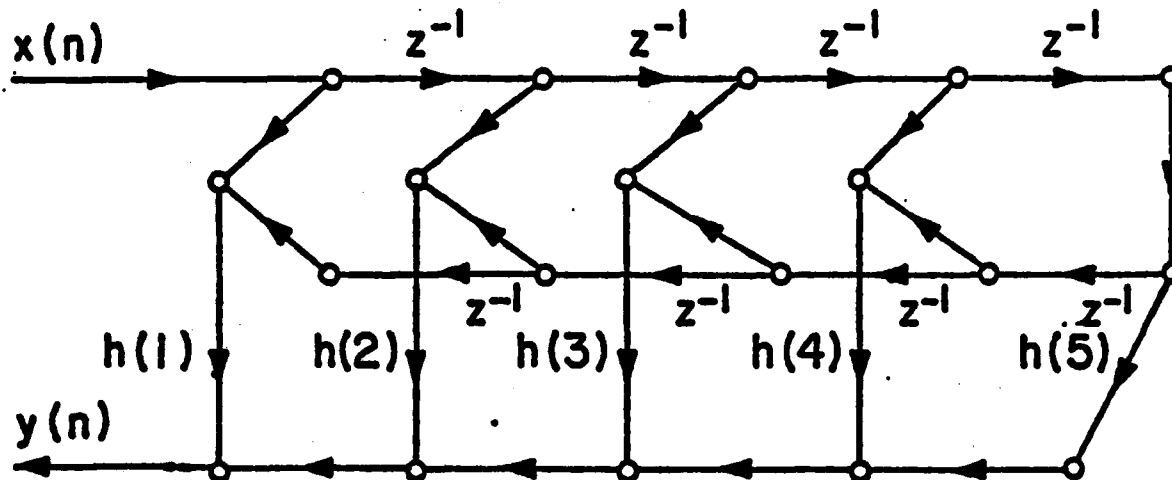
```

IFLAG=0
25 READ(1,15,END=300)(IPICH(I),I=1,16)
KK=16
C*****
C*****
C*****PULSE GENERATION.
C*****
30 DO 50 I=1,KK
    MAG=IPICH(I)
C*****
C*****SET FIRST NP VALUES TO NP, WHERE NP=THE PITCH
C*****
C*****VALUE.
C*****
    DO 100 J=1,MAG
        CNT=CNT+1
        INC=MODN(CNT)
        ISP(INC)=MAG
        IF(INC.EQ.16)WRITE(2,10)(ISP(K),K=1,16)
100 CONTINUE
C*****
C*****SET SECOND NP VALUES TO 0, WHERE NP=THE PITCH
C*****
C*****VALUE.
C*****
    DO 200 J=1,MAG
        CNT=CNT+1
        INC=MODN(CNT)
        ISP(INC)=0
        IF(INC.EQ.16)WRITE(2,10)(ISP(K),K=1,16)
200 CONTINUE
50 CONTINUE
IF(IFLAG.EQ.1)GO TO 400
GO TO 25
300 IFLAG=1
KK=1
GO TO 30
400 CONTINUE
WRITE(2,10)(ISP(K),K=1,INC)
CLOSE(UNIT=1)
CLOSE(UNIT=2)
STOP
END
C
C
C
C
C*****
C*****

```

# APPENDIX D

## FILTER DESIGN FINITE IMPULSE RESPONSE (FIR), LINEAR PHASE, 9TH ORDER, DIGITAL LOW PASS FILTER SPECIFICATIONS



### IMPULSE RESPONSE

|                 | 8 kHz<br>Wc = 1 kHz | 8 kHz<br>Wc = 500 Hz | 6.4 kHz<br>Wc = 1 kHz | 6.4 kHz<br>Wc = 500 Hz |
|-----------------|---------------------|----------------------|-----------------------|------------------------|
| $h(1) = h(9) =$ | 0.056               | 0.034                | 0.071                 | 0.038                  |
| $h(2) = h(8) =$ | 0.072               | 0.013                | 0.155                 | 0.021                  |
| $h(3) = h(7) =$ | 0.101               | 0.014                | 0.234                 | 0.025                  |
| $h(4) = h(6) =$ | 0.122               | 0.015                | 0.319                 | 0.028                  |
| $h(5) = h(5) =$ | 0.130               | 0.015                | 0.336                 | 0.029                  |
|                 | 8k//1k              | 8k//.5k              | 6.4//1k               | 6.4//.5k               |

#### Band 1:

|                 |       |       |       |       |
|-----------------|-------|-------|-------|-------|
| Lower Band Edge | 0.0   | 0.0   | 0.0   | 0.0   |
| Upper Band Edge | 0.125 | 0.063 | 0.156 | 0.078 |

#### Band 2:

|                 |       |       |       |       |
|-----------------|-------|-------|-------|-------|
| Lower Band Edge | 0.131 | 0.069 | 0.164 | 0.086 |
| Upper Band Edge | 0.500 | 0.500 | 0.500 | 0.500 |

**APPENDIX E**  
**FINAL OUTPUT EXAMPLES**

**"MOVE THE VAT OVER THE HOT FIRE"**

**"CATS AND DOGS EACH HATE THE OTHER"**

SENTENCE:

MOVE THE VAT OVER THE HOT FIRE.

ALP= 8.97 ALAD= 8.95 G=8.815 N= 4 RMSMIN= 38.8 SHIN= 8.38 ABIAS=8.81

THE FOLLOWING IS A LIST OF SCALING FACTORS,  
EXPANSION FACTORS, AND THRESHOLDS FOR A 15 LEVEL  
QUANTIZER( 8 ON A SIDE)

| LEVEL NUMBER       | SCALING FACTORS | THRESHOLDS                            | EXPANSION FACTORS |
|--------------------|-----------------|---------------------------------------|-------------------|
| 1                  | 8.8888          | 8.1639                                | 8.5888            |
| 2                  | 8.3277          | 8.5453                                | 8.7588            |
| 3                  | 8.7629          | 1.8517                                | 1.8888            |
| 4                  | 1.3486          | 1.7242                                | 1.8888            |
| 5                  | 2.1877          | 2.6169                                | 1.8888            |
| 6                  | 3.1262          | 3.8823                                | 2.8888            |
| 7                  | 4.4784          | 6.2392                                | 3.8888            |
| 8                  | 8.8888          | 8.8888                                | 5.8888            |
| SNR INLOOP= 26.328 | H=2.91          | OP 1588 1167 1287 738 753 358 316 138 |                   |
|                    | 37 5 32 8       |                                       |                   |

| SAMPLE NUMBER | SNR   | SPER  | SPERI | SNRO  |
|---------------|-------|-------|-------|-------|
| 1- 64         | 24.77 | 12.11 | 13.64 | 12.66 |
| 65- 128       | 27.84 | 13.99 | 15.67 | 13.85 |
| 129- 192      | 28.49 | 14.86 | 15.23 | 14.43 |
| 193- 256      | 28.54 | 14.69 | 15.77 | 13.95 |
| 257- 320      | 38.74 | 15.84 | 16.88 | 15.69 |
| 321- 384      | 38.24 | 16.35 | 17.11 | 13.89 |
| 385- 448      | 28.82 | 14.77 | 17.32 | 13.24 |
| 449- 512      | 27.53 | 14.95 | 17.82 | 12.58 |
| 513- 576      | 28.71 | 14.73 | 16.54 | 13.98 |
| 577- 640      | 24.99 | 12.65 | 15.83 | 12.34 |
| 641- 704      | 25.32 | 11.63 | 12.57 | 13.69 |
| 705- 768      | 24.12 | 11.91 | 12.95 | 12.21 |
| 769- 832      | 28.96 | 18.18 | 18.45 | 18.86 |
| 833- 896      | 21.21 | 11.86 | 13.42 | 9.35  |
| 897- 960      | 26.84 | 9.98  | 9.68  | 16.93 |
| 961- 1024     | 27.98 | 18.58 | 18.96 | 17.48 |
| 1025- 1088    | 24.18 | 8.26  | 9.93  | 15.92 |
| 1089- 1152    | 23.72 | 9.24  | 18.81 | 14.48 |
| 1153- 1216    | 22.91 | 11.28 | 13.27 | 11.62 |
| 1217- 1280    | 28.38 | 18.17 | 12.24 | 18.12 |
| 1281- 1344    | 21.85 | 18.78 | 12.88 | 11.87 |
| 1345- 1408    | 28.38 | 9.38  | 18.29 | 18.92 |

|            |       |       |       |       |
|------------|-------|-------|-------|-------|
| 1489- 1472 | 27.38 | 18.21 | 18.63 | 17.89 |
| 1473- 1536 | 27.91 | 18.85 | 11.83 | 17.85 |
| 1537- 1688 | 26.48 | 18.62 | 11.47 | 15.77 |
| 1681- 1664 | 28.73 | 11.83 | 11.65 | 17.78 |
| 1668- 1728 | 25.91 | 9.19  | 18.25 | 16.72 |
| 1729- 1792 | 25.82 | 9.65  | 18.25 | 16.17 |
| 1793- 1856 | 25.93 | 9.42  | 9.85  | 16.51 |
| 1857- 1928 | 26.12 | 18.22 | 18.86 | 15.98 |
| 1921- 1984 | 25.77 | 18.29 | 11.14 | 15.48 |
| 1985- 2048 | 27.24 | 11.82 | 11.36 | 16.22 |
| 2049- 2112 | 28.38 | 13.83 | 14.68 | 14.47 |
| 2113- 2176 | 17.26 | 7.65  | 11.43 | 9.61  |
| 2177- 2248 | 13.28 | 7.25  | 12.68 | 5.94  |
| 2241- 2304 | 8.76  | 2.63  | 8.36  | 6.13  |
| 2305- 2368 | 18.18 | 9.84  | 9.16  | 9.85  |
| 2369- 2432 | 28.93 | 13.88 | 14.88 | 15.93 |
| 2433- 2496 | 29.95 | 13.28 | 13.69 | 16.67 |
| 2497- 2568 | 38.53 | 14.76 | 15.97 | 15.77 |
| 2561- 2624 | 29.43 | 14.36 | 15.28 | 15.87 |
| 2625- 2688 | 29.72 | 14.74 | 15.89 | 14.98 |
| 2689- 2752 | 29.93 | 15.12 | 16.46 | 14.88 |
| 2753- 2816 | 29.16 | 15.88 | 16.41 | 14.16 |
| 2817- 2888 | 28.66 | 14.69 | 16.98 | 13.97 |
| 2881- 2944 | 27.47 | 14.96 | 16.81 | 12.51 |
| 2945- 3008 | 27.47 | 14.82 | 15.77 | 13.45 |
| 3009- 3072 | 28.96 | 15.23 | 16.43 | 13.73 |
| 3073- 3136 | 38.35 | 14.48 | 15.52 | 15.87 |
| 3137- 3208 | 29.91 | 13.87 | 15.84 | 16.84 |
| 3201- 3264 | 28.18 | 13.83 | 15.18 | 14.27 |
| 3265- 3328 | 28.97 | 14.35 | 15.44 | 14.62 |
| 3329- 3392 | 18.37 | 8.79  | 12.98 | 9.58  |
| 3393- 3456 | 17.98 | 11.91 | 18.18 | 5.99  |
| 3457- 3528 | 11.28 | 4.39  | 7.54  | 6.98  |
| 3521- 3584 | 38.52 | 13.37 | 14.89 | 17.15 |
| 3585- 3648 | 24.83 | 9.42  | 11.51 | 15.41 |
| 3649- 3712 | 26.41 | 18.98 | 12.14 | 15.51 |
| 3713- 3776 | 22.85 | 9.29  | 18.53 | 13.56 |
| 3777- 3848 | 21.28 | 5.92  | 6.75  | 15.28 |
| 3841- 3904 | 21.62 | 4.86  | 5.82  | 16.76 |
| 3905- 3968 | 18.42 | 2.92  | 3.55  | 15.58 |
| 3969- 4032 | 24.75 | 7.48  | 8.18  | 17.27 |
| 4033- 4096 | 24.81 | 11.81 | 12.78 | 12.99 |
| 4097- 4168 | 21.76 | 6.77  | 8.78  | 14.99 |
| 4161- 4224 | 24.66 | 7.49  | 8.98  | 17.17 |
| 4225- 4288 | 25.83 | 8.38  | 8.99  | 16.65 |
| 4289- 4352 | 26.75 | 9.49  | 11.87 | 17.26 |
| 4353- 4416 | 27.71 | 11.68 | 11.38 | 16.11 |
| 4417- 4488 | 17.37 | 9.35  | 18.84 | 8.82  |
| 4481- 4544 | 5.34  | 1.53  | 5.84  | 3.81  |
| 4545- 4608 | 8.18  | -1.71 | 1.89  | 1.98  |
| 4609- 4672 | 5.87  | -1.69 | -8.82 | 7.56  |
| 4673- 4736 | 12.51 | -8.81 | 1.79  | 12.52 |

|           |       |       |       |       |
|-----------|-------|-------|-------|-------|
| 4737-4888 | 15.86 | 1.96  | 3.81  | 13.98 |
| 4881-4864 | 17.25 | 2.67  | 3.38  | 14.58 |
| 4865-4928 | 17.49 | 4.67  | 5.54  | 12.82 |
| 4929-4992 | 17.96 | 4.82  | 5.37  | 13.14 |
| 4993-5056 | 15.68 | 2.51  | 2.84  | 13.18 |
| 5057-5120 | 28.46 | 11.83 | 12.48 | 16.62 |
| 5121-5184 | 27.39 | 11.16 | 12.35 | 16.72 |
| 5185-5248 | 27.12 | 18.83 | 18.78 | 17.89 |
| 5249-5312 | 28.89 | 18.48 | 18.85 | 17.69 |
| 5313-5376 | 23.33 | 7.45  | 7.57  | 15.88 |
| 5377-5440 | 22.43 | 7.99  | 8.32  | 14.44 |
| 5441-5504 | 21.51 | 7.87  | 8.45  | 14.54 |
| 5505-5568 | 26.13 | 9.36  | 9.75  | 16.77 |
| 5569-5632 | 24.88 | 11.81 | 12.83 | 13.87 |
| 5633-5696 | 25.46 | 8.83  | 8.52  | 16.62 |
| 5697-5760 | 28.72 | 5.73  | 7.78  | 14.99 |
| 5761-5824 | 21.75 | 8.13  | 9.58  | 13.61 |
| 5825-5888 | 25.51 | 9.78  | 18.92 | 15.91 |
| 5889-5952 | 28.86 | 9.48  | 11.68 | 11.38 |
| 5953-6016 | 26.28 | 9.92  | 9.73  | 16.36 |
| 6017-6080 | 23.28 | 8.55  | 9.88  | 14.73 |
| 6081-6144 | 16.59 | 6.83  | 8.24  | 18.66 |
| 6145-6208 | 15.26 | 5.34  | 8.15  | 9.93  |
| 6209-6272 | 14.43 | 5.89  | 8.97  | 8.54  |
| 6273-6336 | 5.88  | 8.69  | 5.78  | 5.19  |
| 6337-6400 | 8.88  | 8.88  | 5.88  | 8.88  |
| 6401-6464 | 8.88  | 8.88  | 8.88  | 8.88  |

PREDICTOR PERFORMANCE = 18.47  
 PREDICTOR IDEAL PERFORMANCE = 11.32  
 SIGNAL TO NOISE RATIO = 15.85

| LEVEL NUMBER | PROBABILITY | FREQUENCY |
|--------------|-------------|-----------|
| 1            | 8.2444      | 1588      |
| 2            | 8.1885      | 1167      |
| 3            | 8.1867      | 1287      |
| 4            | 8.1129      | 738       |
| 5            | 8.1165      | 753       |
| 6            | 8.8541      | 358       |
| 7            | 8.8489      | 316       |
| 8            | 8.8281      | 138       |
| 9            | 8.8184      | 119       |
| 10           | 8.8846      | 38        |
| 11           | 8.8857      | 37        |
| 12           | 8.8888      | 5         |
| 13           | 8.8858      | 32        |
| 14           | 8.8888      | 8         |
| 15           | 8.8812      | 3         |
| 16           | 8.8888      | 8         |

SENTENCE:

CATS AND DOGS EACH HATE THE OTHER.

ALP= 0.97 ALAD= 0.96 C=0.015 N= 4 RMSHIN= 35.0 SMIN= 0.30 ABIAS=0.01

THE FOLLOWING IS A LIST OF SCALING FACTORS,  
EXPANSION FACTORS, AND THRESHOLDS FOR A 15 LEVEL  
QUANTIZER( 8 ON A SIDE)

| LEVEL NUMBER       | SCALING FACTORS | THRESHOLDS                            | EXPANSION FACTORS |
|--------------------|-----------------|---------------------------------------|-------------------|
| 1                  | 0.0000          | 0.1639                                | 0.5000            |
| 2                  | 0.3277          | 0.5453                                | 0.7500            |
| 3                  | 0.7029          | 1.0517                                | 1.0000            |
| 4                  | 1.3406          | 1.7242                                | 1.0000            |
| 5                  | 2.1077          | 2.6169                                | 1.0000            |
| 6                  | 3.1262          | 3.8023                                | 2.0000            |
| 7                  | 4.4784          | 6.2392                                | 3.0000            |
| 8                  | 0.0000          | 0.0000                                | 5.0000            |
| SNR INLOOP= 23.152 | H=3.17          | OP 1973 1082 1137 865 843 618 604 382 |                   |
|                    | 120 42 40       | 1 9                                   |                   |

| SAMPLE NUMBER | SNR   | SPER  | SPERI | SNRQ  |
|---------------|-------|-------|-------|-------|
| 1- 64         | 17.67 | 0.18  | -0.07 | 17.50 |
| 65- 128       | 11.62 | -4.33 | -5.60 | 15.95 |
| 129- 192      | 20.89 | 3.76  | 4.63  | 17.13 |
| 193- 256      | 24.73 | 4.66  | 4.38  | 20.07 |
| 257- 320      | 23.77 | 7.46  | 7.86  | 16.31 |
| 321- 384      | 25.45 | 7.96  | 8.02  | 17.50 |
| 385- 448      | 18.74 | 2.91  | 4.08  | 15.83 |
| 449- 512      | 21.11 | 3.59  | 2.80  | 17.52 |
| 513- 576      | 19.44 | 4.22  | 3.16  | 15.21 |
| 577- 640      | 19.27 | 2.39  | 3.27  | 16.89 |
| 641- 704      | 23.32 | 5.22  | 4.96  | 18.10 |
| 705- 768      | 15.78 | 9.90  | 13.78 | 5.00  |
| 769- 832      | 4.25  | -0.21 | 4.41  | 4.45  |
| 833- 896      | 8.43  | -3.53 | -3.98 | 11.96 |
| 897- 960      | 3.77  | -5.04 | -4.88 | 8.01  |
| 961- 1024     | 7.69  | -4.67 | -5.65 | 12.36 |
| 1025- 1088    | 8.23  | -4.26 | -5.94 | 12.49 |
| 1089- 1152    | 12.51 | -3.18 | -4.88 | 15.69 |
| 1153- 1216    | 10.45 | -5.57 | -6.77 | 16.03 |
| 1217- 1280    | 9.61  | -5.00 | -6.09 | 14.62 |
| 1281- 1344    | 24.41 | 8.37  | 9.18  | 16.04 |
| 1345- 1408    | 24.19 | 7.23  | 7.15  | 16.97 |

|       |      |       |       |       |       |
|-------|------|-------|-------|-------|-------|
| 1469- | 1472 | 21.69 | 5.95  | 6.38  | 15.74 |
| 1473- | 1536 | 23.77 | 8.68  | 8.88  | 15.89 |
| 1537- | 1608 | 23.42 | 8.38  | 9.18  | 15.12 |
| 1601- | 1664 | 25.58 | 18.28 | 18.89 | 15.38 |
| 1665- | 1728 | 25.65 | 18.22 | 9.59  | 15.43 |
| 1729- | 1792 | 25.58 | 9.78  | 9.85  | 15.88 |
| 1793- | 1856 | 25.55 | 9.88  | 9.98  | 15.75 |
| 1857- | 1928 | 24.87 | 9.12  | 8.86  | 15.75 |
| 1921- | 1984 | 25.25 | 14.28 | 15.71 | 11.86 |
| 1985- | 2048 | 25.23 | 16.88 | 18.72 | 9.14  |
| 2049- | 2112 | 22.11 | 13.23 | 16.47 | 8.88  |
| 2113- | 2176 | 17.72 | 1.19  | -8.18 | 16.53 |
| 2177- | 2248 | 24.85 | 8.47  | 7.73  | 15.58 |
| 2241- | 2314 | 26.55 | 9.93  | 18.91 | 16.62 |
| 2315- | 2368 | 28.87 | 13.78 | 14.48 | 15.89 |
| 2369- | 2432 | 26.99 | 12.89 | 14.38 | 14.18 |
| 2433- | 2496 | 29.23 | 13.64 | 14.59 | 15.59 |
| 2497- | 2568 | 28.65 | 13.68 | 14.24 | 14.97 |
| 2561- | 2624 | 27.35 | 11.82 | 12.71 | 15.53 |
| 2625- | 2688 | 28.68 | 12.74 | 12.69 | 15.85 |
| 2689- | 2752 | 26.73 | 18.51 | 12.49 | 16.22 |
| 2753- | 2816 | 26.26 | 9.59  | 18.51 | 16.67 |
| 2817- | 2888 | 28.33 | 13.18 | 15.15 | 15.15 |
| 2881- | 2944 | 27.57 | 11.78 | 12.78 | 15.79 |
| 2945- | 3008 | 29.89 | 13.89 | 13.38 | 16.88 |
| 3009- | 3072 | 27.57 | 18.98 | 11.48 | 16.59 |
| 3073- | 3136 | 25.89 | 12.28 | 12.75 | 13.61 |
| 3137- | 3208 | 22.26 | 18.91 | 12.26 | 11.35 |
| 3201- | 3264 | 26.17 | 13.47 | 14.72 | 12.78 |
| 3265- | 3328 | 25.58 | 11.82 | 13.27 | 13.68 |
| 3329- | 3392 | 28.52 | 11.23 | 13.61 | 9.38  |
| 3393- | 3456 | 19.89 | 5.66  | 7.88  | 14.23 |
| 3457- | 3528 | 17.65 | 1.98  | 1.58  | 15.67 |
| 3521- | 3584 | 11.95 | -4.52 | -4.86 | 16.47 |
| 3585- | 3648 | 18.17 | -4.68 | -5.89 | 14.85 |
| 3649- | 3712 | 8.57  | -7.89 | -8.25 | 16.46 |
| 3713- | 3776 | 18.65 | -4.88 | -5.86 | 15.45 |
| 3777- | 3848 | 8.46  | -7.36 | -8.61 | 15.82 |
| 3841- | 3904 | 18.61 | -7.88 | -8.81 | 17.69 |
| 3905- | 3968 | 6.98  | -8.91 | -9.23 | 15.88 |
| 3969- | 4032 | 6.68  | -9.54 | -8.86 | 16.14 |
| 4033- | 4096 | 9.86  | -5.57 | -6.15 | 15.43 |
| 4097- | 4168 | 8.38  | -6.82 | -6.82 | 14.32 |
| 4161- | 4224 | 3.57  | -5.34 | -2.89 | 8.91  |
| 4225- | 4288 | -8.31 | -8.31 | -8.59 | 8.88  |
| 4289- | 4352 | 8.88  | 8.88  | -1.91 | 8.88  |
| 4353- | 4416 | 8.88  | 8.88  | -2.83 | 8.88  |
| 4417- | 4488 | 8.88  | 8.88  | -3.81 | 8.88  |
| 4481- | 4544 | 8.88  | 8.88  | -3.32 | 8.88  |
| 4545- | 4608 | 8.88  | 8.88  | -6.86 | 8.88  |
| 4609- | 4672 | 9.59  | 6.84  | 9.28  | 3.54  |
| 4673- | 4736 | 17.52 | 3.35  | 3.33  | 14.17 |

|           |       |        |        |       |
|-----------|-------|--------|--------|-------|
| 4737-4888 | 18.98 | 2.11   | 2.81   | 16.79 |
| 4881-4864 | 18.58 | 1.88   | 8.82   | 16.62 |
| 4865-4928 | 16.72 | -8.21  | 8.26   | 16.93 |
| 4929-4992 | 17.68 | 8.84   | 8.28   | 17.64 |
| 4993-5056 | 13.27 | -3.68  | -3.15  | 16.86 |
| 5057-5120 | 16.76 | -8.14  | 1.24   | 16.89 |
| 5121-5184 | 13.11 | 5.36   | 7.81   | 7.76  |
| 5185-5248 | 8.88  | 8.88   | -8.81  | 8.88  |
| 5249-5312 | 9.45  | -7.51  | -7.18  | 16.96 |
| 5313-5376 | 8.75  | -9.29  | -9.83  | 18.84 |
| 5377-5440 | 7.88  | -18.27 | -18.57 | 17.36 |
| 5441-5504 | 8.26  | -8.55  | -9.92  | 16.81 |
| 5505-5568 | 9.98  | -5.69  | -6.34  | 15.59 |
| 5569-5632 | 15.18 | -1.88  | -1.68  | 16.97 |
| 5633-5696 | 26.46 | 6.41   | 7.52   | 28.85 |
| 5697-5760 | 19.81 | 2.89   | 2.85   | 16.92 |
| 5761-5824 | 19.58 | 1.58   | 8.91   | 17.92 |
| 5825-5888 | 15.81 | 8.81   | 8.81   | 15.81 |
| 5889-5952 | 15.98 | -8.86  | -8.68  | 16.76 |
| 5953-6016 | 15.18 | -1.32  | -2.26  | 16.49 |
| 6017-6080 | 28.87 | 7.73   | 6.92   | 13.14 |
| 6081-6144 | 5.49  | 8.39   | 3.47   | 5.18  |
| 6145-6208 | 8.88  | 8.88   | 1.39   | 8.88  |
| 6209-6272 | 8.88  | 8.88   | -4.81  | 8.88  |
| 6273-6336 | 8.88  | 8.88   | -5.79  | 8.88  |
| 6337-6400 | 17.48 | 2.81   | 1.24   | 14.59 |
| 6401-6464 | 17.27 | 4.31   | 4.85   | 12.97 |
| 6465-6528 | 15.98 | -8.86  | 8.26   | 16.84 |
| 6529-6592 | 14.78 | -8.68  | -2.86  | 15.38 |
| 6593-6656 | 21.78 | 5.84   | 3.92   | 16.65 |
| 6657-6720 | 16.21 | 8.38   | 1.35   | 15.92 |
| 6721-6784 | 14.87 | 8.27   | 8.72   | 14.59 |
| 6785-6848 | 14.55 | -3.11  | -2.44  | 17.66 |
| 6849-6912 | 24.98 | 8.18   | 7.99   | 16.88 |
| 6913-6976 | 24.33 | 8.63   | 18.82  | 15.78 |
| 6977-7040 | 27.63 | 12.15  | 11.95  | 15.48 |
| 7041-7104 | 38.29 | 12.95  | 13.76  | 17.35 |
| 7105-7168 | 27.67 | 11.17  | 18.68  | 16.58 |
| 7169-7232 | 28.88 | 9.92   | 18.88  | 18.16 |
| 7233-7296 | 26.55 | 11.57  | 11.69  | 14.98 |
| 7297-7360 | 26.88 | 15.23  | 16.82  | 11.65 |
| 7361-7424 | 24.85 | 11.42  | 11.29  | 12.63 |
| 7425-7488 | 21.89 | 6.45   | 7.46   | 15.44 |
| 7489-7552 | 23.88 | 8.36   | 8.86   | 14.64 |
| 7553-7616 | 24.48 | 6.21   | 6.68   | 18.19 |
| 7617-7680 | 23.84 | 8.81   | 8.93   | 14.23 |
| 7681-7744 | 22.38 | 6.47   | 6.57   | 15.91 |
| 7745-7808 | 21.88 | 11.14  | 12.93  | 9.94  |
| 7809-7872 | 14.81 | 6.48   | 8.39   | 8.33  |
| 7873-7936 | 16.15 | 3.58   | 4.86   | 12.65 |
| 7937-8000 | 3.83  | -1.69  | 7.43   | 5.52  |
| 8001-8064 | 8.88  | 8.88   | 8.88   | 8.88  |

PREDICTOR PERFORMANCE = 6.66  
 PREDICTOR IDEAL PERFORMANCE = 6.81  
 SIGNAL TO NOISE RATIO = 16.49

| LEVEL NUMBER | PROBABILITY | FREQUENCY |
|--------------|-------------|-----------|
| 1            | 0.2438      | 1973      |
| 2            | 0.1337      | 1082      |
| 3            | 0.1495      | 1137      |
| 4            | 0.1069      | 865       |
| 5            | 0.1042      | 843       |
| 6            | 0.0764      | 618       |
| 7            | 0.0746      | 604       |
| 8            | 0.0373      | 302       |
| 9            | 0.0410      | 332       |
| 10           | 0.0153      | 124       |
| 11           | 0.0148      | 120       |
| 12           | 0.0052      | 42        |
| 13           | 0.0049      | 40        |
| 14           | 0.0001      | 1         |
| 15           | 0.0011      | 9         |
| 16           | 0.0000      | 0         |
| 17           | 0.0000      | 0         |
| 18           | 0.0000      | 0         |
| 19           | 0.0000      | 0         |
| 20           | 0.0000      | 0         |
| 21           | 0.0000      | 0         |
| 22           | 0.0000      | 0         |
| 23           | 0.0000      | 0         |
| 24           | 0.0000      | 0         |
| 25           | 0.0000      | 0         |
| 26           | 0.0000      | 0         |
| 27           | 0.0000      | 0         |
| 28           | 0.0000      | 0         |
| 29           | 0.0000      | 0         |
| 30           | 0.0000      | 0         |
| 31           | 0.0000      | 0         |
| 32           | 0.0000      | 0         |

## REFERENCES

1. Arora, A.S., Melsa, J.L., Mills, J.D., "Real Time Implementation of a Speech Coding Algorithm Using Time-Domain Harmonic Scaling and Adaptive Residual Coder", Proc. 1983 IEEE Int. Conf. Acous., Speech, Signal Processing, April 1983.
2. Cohn, D.L., Melsa, J.L., "The Relationship Between an Adaptive Quantizer and a Variance Estimator", IEEE Trans. Inform. Theory, Vol. IT-21, Nov. 1975.
3. Cohn, D.L., Melsa, J.L., "The Residual Encoder and Improved ADPCM System for Speech Digitization", IEEE Trans. Commun., Vol. COM-23, September 1975.
4. Cox, R.V., Crochiere, R.E., Malah, D., "Performance of Transform and Subband Coding Systems Combined with Harmonic Scaling of Speech", IEEE Trans. on Acous. Speech and Signal Processing, Vol. ASSP-29, No. 2, April 1981.
5. Dubnowski, J.J., Rabiner, L.R., Schafer, R.W., "Real-Time Digital Hardware Pitch Detector", IEEE Trans. on Acous. Speech and Signal Processing, Vol. ASSP-24, No. 1, February 1976.
6. Luenberger, D.G., Introduction to Linear and Nonlinear Programming, Wesley Publishing Company, Massachusetts, 1965.
7. Defense Communication Agency, "Statement of Work: Contract Number DCA R610-82-029", December 1981.
8. Malah, D., "Combined Time-Domain Harmonic Compression and DVSD for 7.2 kbit/s Transmission of Speech Signals", Proc. 1980 IEEE Int. Conf. Acous. Speech, Signal Processing, April 1980.
9. Malah, D., "Time-Domain Algorithms for Harmonic Bandwidth Reduction and Time Scaling of Speech Signals", IEEE Trans. on ASSP, Vol. ASSP-27, No. 2, April 1979.
10. McDonald, R.A., "Signal-to-Noise and Idle Channel Performance of Differential Pulse Code Modulation Systems - Particular Applications to Voice Signals", Bell Syst. Tech. J., April 1966.
11. Melsa, James L., "Unsolicited Proposal to the Defense Communications Agency for support of a work entitled Real-Time Implementation of a Speech Digitization Algorithm Combining Time-Domain Harmonic Scaling and Adaptive Residual Coding", August 1981.
12. Melsa, J.L., Cohn, D.L., Kresse, J., Arora, A., Pande, A., "A Waveform Encoder for Speech Transmission Over 9.6 kb/s Noisy Channels", NTC Conference Record, Vol. 3, No. 50.3, December 1980.

13. Melsa, J.L., Pande, A.K., "Mediumband Speech Encoding Using Time Domain Harmonic Scaling and Adaptive Residual Coding", Proc. 1981 Int. Conf. ASSP, Atlanta, GA, April 1981.
14. Melsa, J.L., Pande, A.K. "A Study of a New Approach to Digital Speech Digitization Combining Time-Domain Harmonic Scaling and Adaptive Residual Coding", Final Report to the Defense Communication Agency, August 1981.
15. Rabiner, L.R., Sambur, M.R. Schmidt, C.E., "Applications of a Nonlinear Smoothing Algorithm to Speech Processing", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-23, No. 6, December 1975.
16. Schafer, R.W., Oppenheim, A.V., Digital Signal Processing, Prentice-Hall, New Jersey, 1975.

END